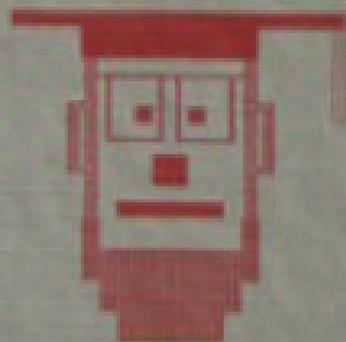




DIGITAL  
RESEARCH®



**Dr. Logo™**  
**Benutzerhandbuch**

Für Schneider CPC 464,664,6128 und Joyce  
Lexikon und Schulungsprogramm.



# Dr. Logo™ Benutzerhandbuch

Für Schneider CPC 464, 664, 6128  
und Joyce.  
Lexikon und Schulungsprogramm.

Originaldokumentation  
Markt & Technik Verlag

CIP-Kurztitelaufnahme der Deutschen Bibliothek

**[Doktor-Logo-Benutzerhandbuch]**

Dr.-Logo-Benutzerhandbuch : für Schneider CPC 464, 664, 6128 u. Joyce ;  
Lexikon u. Schulungsprogramm / Digital Research. -  
Haar bei München : Markt-und-Technik-Verlag, 1986.  
(Markt&Technik Originaldokumentation)  
ISBN 3-89090-116-6

NE: Digital Research <Pacific Grove, Calif.>

Die Informationen im vorliegenden Buch werden ohne Rücksicht auf einen eventuellen Patentschutz veröffentlicht.  
Warennamen werden ohne Gewährleistung der freien Verwendbarkeit benutzt.  
Bei der Zusammenstellung von Texten und Abbildungen wurde mit größter Sorgfalt vorgegangen.  
Trotzdem können Fehler nicht vollständig ausgeschlossen werden.  
Verlag, Herausgeber und Autoren können für fehlerhafte Angaben und deren Folgen weder eine juristische  
Verantwortung noch irgendeine Haftung übernehmen.  
Für Verbesserungsvorschläge und Hinweise auf Fehler sind Verlag und Herausgeber dankbar.

Alle Rechte vorbehalten, auch die der fotomechanischen Wiedergabe und der Speicherung in elektronischen Medien.  
Die gewerbliche Nutzung der in diesem Buch gezeigten Modelle und Arbeiten ist nicht zulässig.

CBASIC, CP/M und CP/NET sind eingetragene Warenzeichen der Digital Research Inc., USA.  
ASM, CP/M Plus, DESPOOL, DDT, LIB-80, LINK-80, MAC, MP/M, MP/M II, Pascal /MT+, PL/I-80, RMAC,  
SID und TEX-80 sind Warenzeichen der Digital Research Inc., USA.  
Intel ist ein eingetragenes Warenzeichen der Intel Corp., USA.  
MicroSoft ist ein eingetragenes Warenzeichen der MicroSoft Corp., USA.  
TI Silent 700 ist ein Warenzeichen der Texas Instruments Inc., USA.  
Zilog und Z80 sind eingetragene Warenzeichen der Zilog Inc., USA.  
Dr. Logo ist ein eingetragenes Warenzeichen der Digital Research Inc., USA.

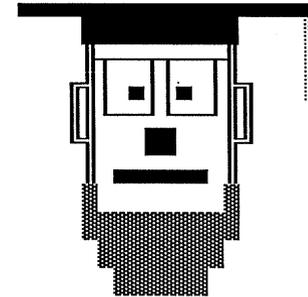
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1  
89 88 87 86

ISBN 3-89090-116-6

© 1986 by Markt&Technik, 8013 Haar bei München  
Alle Rechte vorbehalten  
Einbandgestaltung: Grafikdesign Heinz Rauner  
Druck: Jantsch, Günzburg  
Printed in Germany



# Gestatten, Dr. Logo!



## Schulungsprogramm

## COPYRIGHT

Copyright 1984 (C) durch Digital Research Inc. Alle Rechte vorbehalten. Dieses Handbuch darf weder teilweise noch als ganzes in irgendeiner Form oder auf irgendeine Weise, sei dies elektronisch, mechanisch, magnetisch, optisch, chemisch, manuell oder auf irgendeine andere Art, reproduziert, übertragen, kopiert, in einem Informationssystem gespeichert oder in irgendeine Sprache bzw. Computersprache übersetzt werden, ohne die vorherige schriftliche Genehmigung von Digital Research, Post Office Box 579, Pacific Grove, California, 93950.

Die Leser dieses Handbuches dürfen die hierin enthaltenen Beispielprogramme ganz oder teilweise in eigenen Programmen verwenden.

## HAFTUNGSAUSSCHLUSS

Digital Research übernimmt keinerlei Haftung oder Gewährleistung in Bezug auf den Inhalt des vorliegenden Handbuches und schließt insbesondere jede stillschweigend mit inbegriffene Garantie hinsichtlich der handelsüblichen Qualität oder der Eignung für irgendwelche bestimmte Zwecke aus. Ferner behält sich Digital Research das Recht vor, das vorliegende Handbuch zu überarbeiten und zu gegebener Zeit dessen Inhalt zu ändern, ohne daß Digital Research verpflichtet ist, irgend jemanden über eine derartige Überarbeitung oder derartige Änderungen zu benachrichtigen.

## WARENZEICHEN

Digital Research und deren Firmenzeichen sind eingetragene Warenzeichen der Firma Digital Research Inc. Das Dr. Logo-Zeichen und Dr. Logo sind Warenzeichen von Digital Research Inc.

Erste Ausgabe: Juni 1984

## Inhaltsverzeichnis

<b>1</b>	<b>Allgemeine Grundlagen</b>	<b>7</b>
1.1	Was ist Dr. Logo?	7
1.2	Wie das vorliegende Handbuch zu verwenden ist	8
1.3	Dr. Logo und die Schneider-Computer	9
1.4	Mit der Tastatur experimentieren	11
<b>2</b>	<b>Der Dialog mit Dr. Logo</b>	<b>15</b>
2.1	Was ist ein Kommando	15
2.2	Zeichnungen mit den Igel-Grafik-Kommandos erstellen	15
2.3	Zusammenfassung	25
<b>3</b>	<b>Neue Prozeduren einfügen</b>	<b>27</b>
3.1	Eine Prozedur schreiben	27
3.2	Namen für Prozeduren wählen	30
3.3	Zusammenfassung	31
<b>4</b>	<b>Prozeduren ändern</b>	<b>33</b>
4.1	Warum man eine Prozedurdefinition ändern können sollte	33
4.2	Wie der Texteditor zu verwenden ist	33
4.3	Die Prozedur funf editieren	35
4.4	Zusammenfassung	37
<b>5</b>	<b>Prozeduren sichern</b>	<b>39</b>
5.1	Wie Daten gespeichert werden	39
5.2	Der Arbeitsspeicher	42
5.3	Wie Informationen gelöscht werden	42
5.4	Zusammenfassung	43
<b>6</b>	<b>Prozeduren mit Eingaben</b>	<b>45</b>
6.1	Wie man Eingaben verwendet	45
6.2	Prozeduren für Zeichnungen verschiedener Grösse	47
6.3	Zusammenfassung	49
<b>7</b>	<b>Programme erstellen</b>	<b>51</b>
7.1	Was sind Ausgaben	51
7.2	Eine Prozedur unter Verwendung anderer Prozeduren schreiben	52
7.3	Zusammenfassung	54

<b>8</b>	<b>Prozeduren, die sich selbst aufrufen</b>	<b>57</b>
8.1	Eine Prozedur schreiben, die sich selbst aufruft	57
8.2	Wie Prozeduren beendet werden	59
8.3	Ein Balkendiagramm zeichnen	61
8.4	Zusammenfassung	64
<b>9</b>	<b>Die Verwendung von Wörtern</b>	<b>67</b>
9.1	Wörter, die nur Wörter sind	67
9.2	Dinge benennen	70
9.3	Ein Telefonverzeichnis anlegen	72
9.4	Merkmalliste	73
9.5	Erweitern des Telefonverzeichnisses unter Verwendung von Merkmallisten	75
9.6	Zusammenfassung	75
<b>10</b>	<b>Wortspiele</b>	<b>77</b>
10.1	Wie Sätze zur Anzeige gebracht werden	77
10.2	Ein Spiel erstellen, das lustige Geschichten anzeigt	79
10.3	Prozeduren, die eine Reaktion von Ihnen erwarten und darauf wiederum antworten	83
10.4	Zusammenfassung	86
<b>11</b>	<b>Fehler suchen und beseitigen</b>	<b>87</b>
11.1	Beobachten Sie Prozeduren bei der Ausführung	87
11.2	Eine Prozedur schreiben und Fehler beseitigen	88
11.3	Zusammenfassung	90

# 1 Allgemeine Grundlagen

Willkommen in der Welt von Dr. Logo! In der nun folgenden Lektion werden Sie in die Programmiersprache Dr. Logo eingeführt, und es wird Ihnen die Handhabung des vorliegenden Handbuches erläutert. Es wird auch erklärt, wie Dr. Logo gestartet wird und wie die Tastatur zu verwenden ist.

## Lernziele

- Sie sollen lernen, was Dr. Logo ist.
- Sie sollen lernen, wie das vorliegende Handbuch zu verwenden ist.
- Sie sollen Dr. Logo auf Ihrem Computer starten.
- Sie sollen mit der Tastatur experimentieren.

## 1.1 Was ist Dr. Logo?

Dr. Logo ist eine einfach zu handhabende Programmiersprache, die ursprünglich dazu ausgelegt war, Kindern den Umgang mit dem Computer beizubringen. Inzwischen ist Dr. Logo mit Hilfe leistungsfähiger Personal Computer auch eine flexible und leistungsfähige Programmiersprache für Erwachsene. Dr. Logo ist für kommerziellen Einsatz, Hobby und ernsthafte Programmierung gleichermaßen interessant, aus den gleichen Gründen, aus denen Kinder davon begeistert waren: Dr. Logo arbeitet im Dialogverkehr und ist sehr einfach zu handhaben.

Sie können sich den Computer-einfach als Ihren Schüler vorstellen und Dr. Logo als die Sprache, in der Sie Ihrem Computer erklären, was er tun soll.

Im Vergleich mit der gesprochenen Sprache verfügt Dr. Logo nur über einen kleinen Wortschatz. Die meisten Wörter bedeuten Tätigkeiten; d.h. wenn Sie diese auf der Tastatur eintippen, macht Ihr Computer etwas.

Wenn Sie Dr. Logo erhalten, versteht er bereits etwa 200 Wörter, die *Grundwörter*, *Kernbefehle* oder auch *Primitive* genannt werden. Einige Grundwörter beziehen sich auf arithmetische Operationen, wie z.B. *sin* und *quotient*; andere sind für die Richtung in grafischen Entwürfen vorgesehen, wie z.B. *fd* und *rt*; und andere wiederum beziehen sich auf Operationen, die durchgeführt werden sollen nach Informationen, die man selbst angibt, wie z.B. *list* und *repeat*. Sie können durch Verwendung dieser Grundwörter auch neue Wörter in den Wortschatz von Dr. Logo aufnehmen (*Prozeduren* genannt). Sie können auch andere Wörter mit Hilfe der von Ihnen bereits definierten Wörter neu aufnehmen!

Ein auf eine menschliche Handlung bezogenes Beispiel für die Verwendung von Grundwörtern zur Bildung einer Prozedur ist *schwimmen*. Stellen Sie sich die Tätigkeiten treten, drehen, ziehen und atmen als die Grundwörter vor und die Prozedur *schwimmen* als eine besondere Kombination dieser Grundwörter. Wenn Sie jemandem erst einmal erklärt haben, was das Wort *schwimmen* bedeutet, weiß er auch, daß diese nur aus einem Wort bestehende Anweisung treten, drehen, ziehen und atmen bedeutet.

Mit den Grundwörtern und den neuen, von Ihnen definierten Prozeduren können Sie Dr. Logo anweisen,

- o Grafiken zu zeichnen
- o schwierige grafische Darstellungen zu erstellen
- o Wortspiele auszuführen
- o ein Verzeichnis von Namen oder Zahlen anzulegen
- o Zahlen grafisch darzustellen

Es gibt noch sehr viele andere Aufgaben, die Sie Dr. Logo später entsprechend Ihren eigenen Bedürfnissen beibringen werden.

Dr. Logo ist zum Teil deshalb so reizvoll, weil Sie ihn unmittelbar anweisen können, den Computer einfache Prozeduren ausführen zu lassen; es gibt jedoch keine Grenzen für komplexere Prozeduren, die Sie selbst erstellen können.

## 1.2 Wie das vorliegende Handbuch zu verwenden ist

Sinn und Zweck des Handbuches ist, Sie in den Umfang der Programmiermöglichkeiten mit Dr. Logo einzuführen. Dies schließt Grafiken mit dem Igel, mathematische Aufgaben und das Arbeiten mit Wörtern ein. Sie werden zwar kein Experte auf einem dieser Gebiete sein, wenn Sie das vorliegende Handbuch durchgearbeitet haben, aber Sie werden genug darüber wissen, um unter Verwendung des *Dr.-Logo-Lexikon* und Ihrer eigenen Vorstellungskraft weitere Schritte zu wagen.

In der Tat ist experimentieren vielleicht der beste Weg, um Dr. Logo kennenzulernen. Wenn Sie während des Durcharbeitens eine Idee bekommen, probieren Sie sie gleich aus! Es ist unmöglich, Dr. Logo zu beschädigen, und wenn Sie einen Fehler machen oder Dr. Logo anweisen, daß er etwas tun soll, von dem er nicht weiß, wie er es tun soll, so gibt er eine Meldung aus, in der das Problem dargestellt wird. Die Dr.-Logo-Figur taucht in diesem Handbuch in Abständen immer wieder auf, und zwar im eigens dafür vorgesehenen *Dr. Logo's Leitfaden*. Damit werden Sie auf den richtigen Weg zurückgeführt, wenn Sie einen Fehler gemacht haben.

So, nun sollen Sie lernen, wie Sie Dr. Logo auf Ihrem Computer laden.

## 1.3 Dr. Logo und die Schneider-Computer

Ihr Computer besteht aus vier wesentlichen Teilen: einem (Farb-)Bildschirm, einem Diskettenlaufwerk, einer Tastatur und einer Einheit, die auch den Zentralprozessor mit einschließt.

Zunächst schalten Sie Ihren Computer ein. Legen Sie anschließend die Diskette mit dem Betriebssystem in Laufwerk A und laden das Programm in den Arbeitsspeicher.

- o Beim **Joyce** müssen Sie dazu nur die Seite 2 der CP/M-Plus-Diskette ins Laufwerk legen und das System wird automatisch geladen.
- o Beim **CPC6128** legen Sie dagegen die Seite 1 der CP/M-Plus-Diskette ins Laufwerk, tippen auf der Tastatur

```
!cpm
```

und schließen die Eingabe mit Return ab.

Sie hören dann surrende Geräusche, und mehrere Meldungen erscheinen auf dem Bildschirm. Nach ein paar Sekunden erscheint folgende Meldung:

```
A>
```

Diese Meldung wird *System-Prompt* (Eingabe-Aufforderungszeichen des Systems) genannt.

Ehe Sie Dr. Logo laden, sollten Sie eine Sicherungskopie der Diskette mit Dr. Logo machen. Dadurch ist deren Inhalt gegen Verlust oder Beschädigung geschützt. Zur Erstellung einer Sicherungskopie müssen Sie folgende Anweisungen beachten:

1. Nehmen Sie eine saubere, leere Diskette, auf die Sie die Dr.-Logo-Originaldiskette kopieren wollen. Wenn die neue Diskette noch nicht initialisiert ist, müssen Sie sie zunächst formatieren, so daß Sie Daten darauf schreiben und diese wieder lesen können.

**Anmerkung:** In Lektion 5 wird erklärt, wie Daten auf einer zweiten Diskette abgespeichert werden. Sie sollten daher jetzt wenigstens Disketten formatieren; eine für die Sicherungskopie von Dr. Logo und eine für die Speicherung Ihrer Dateien in Lektion 5.

Vergewissern Sie sich, daß Sie auf die Diskette schreiben und sie wieder lesen können und sie nicht schreibgeschützt ist: überprüfen Sie, ob die Schieber an der rechten oberen Ecke der Diskette geschlossen sind bzw. die Steller nach außen geschoben sind.

Verwenden Sie zum Formatieren der Disketten beim CPC6128 das Formatierprogramm DISCKIT3 bzw. DISCKIT beim Joyce. Denken Sie daran, daß das Initialisieren einer Diskette bereits darauf vorhandene Daten löscht.

- Benutzen Sie wiederum die Hilfsprogramme DISCKIT3 beim CPC6128 bzw. DISCKIT beim Joyce, um den Inhalt der Dr.-Logo-Diskette (Seite 3 bzw. Seite 4 der Systemdisketten) zu kopieren, wenn Sie eine Sicherungskopie machen.

Jetzt haben Sie eine Sicherungskopie von Dr.-Logo und wenigstens eine weitere formatierte Diskette. Bewahren Sie die Originaldiskette mit Dr.-Logo an einem sicheren Ort auf. Verwenden Sie diese nur, wenn Sie eine weitere Sicherungskopie erstellen wollen.

- Legen Sie jetzt die Sicherungskopie, *Arbeitsdiskette* genannt, in das Systemlaufwerk ein und löschen mit dem Befehl

```
era help.*
```

die nicht benötigten Hilfe-Dateien. Wenn Sie Besitzer eines CPC6128 sind, haben Sie nun eine (hoffentlich) funktionsfähige Arbeitsdiskette. Wenn Sie jedoch eine Joyce haben, müssen Sie noch die drei Hilfsprogramme SUBMIT.COM, LANGUAGE.COM und SETKEYS.COM auf die Arbeitsdiskette kopieren. Legen Sie dazu die zweite Seite der CP/M-Plus-Diskette ein und rufen das PIP-Kopierprogramm auf:

```
A>pip
*
```

Nach dem Prompt, dem Sternchen, geben Sie nacheinander die folgenden drei Zeilen ein, die Sie jeweils mit Return abschließen:

```
b:=a:submit.com
b:=a:language.com
b:=a:setkeys.com
```

Falls Sie nur ein Diskettenlaufwerk besitzen, müssen Sie mehrmals auf Anforderung die Dr.-Logo-Arbeitsdiskette mit der Systemdiskette tauschen. Haben Sie dagegen zwei Laufwerke, dann legen Sie die Arbeitsdiskette in das zweite ein.

Wenn die Kopiervorgänge ohne Fehlermeldungen abgelaufen sind, können Sie die soeben erstellte Arbeitsdiskette in Laufwerk A: einlegen.

Auf dem CPC6128 starten Sie Dr. Logo nun, indem Sie nach dem System-Prompt

```
A>submit logo3
```

eintippen und mit Return abschicken.

Auf dem Joyce starten Sie Dr. Logo, indem Sie nach dem System-Prompt

```
A>submit logo
```

eintippen und mit Return abschicken.

**Anmerkung:** Wenn Sie Dr. Logo in dieser Weise auf dem Joyce starten, sehen Sie auf dem Bildschirm statt der eckigen Klammern und des umgekehrten Schrägstriches die Großbuchstaben Ä (À) und Ú (U) bzw. Ö (O). Sie haben dafür den Vorteil, daß Sie mit Umlauten arbeiten können - jedoch nur mit den Kleinbuchstaben, da Dr. Logo die großen ja anders versteht. Wenn Sie auf die kleinen Umlaute verzichten wollen, laden sie vor dem Programmstart durch Eingabe von language 0 den amerikanischen Zeichensatz. Bei den Beispielen in den Übungen und im Lexikonteil werden die Umlaute jedoch genutzt.

Sie hören nun wieder surrende Geräusche, und mehrere Meldungen erscheinen auf Ihrem Bildschirm. Nach ein paar Sekunden werden die folgenden Zeichen ausgegeben:

```
?■
```

Das Fragezeichen in der linken oberen Ecke des Bildschirms ist das Prompt von Dr. Logo. Sie können es sich so vorstellen, daß Dr. Logo fragt: *Was jetzt?*

Wenn Sie Schwierigkeiten mit dem Starten von Dr. Logo haben, überprüfen Sie noch einmal, ob Sie die Anweisungen zum Kopieren der Originaldiskette genau befolgt haben. Wenn dies der Fall ist und Sie nach wie vor Schwierigkeiten haben, sehen Sie in Kapitel 1 des Dr.-Logo-Lexikons nach, wo Sie weitere Einzelheiten über das Starten von Dr. Logo finden.

## 1.4 Mit der Tastatur experimentieren

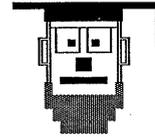
Die Tastatur eines Computers besteht aus alphanumerischen Tasten im mittleren Teil der Tastatur. Dr. Logo kennt den Unterschied zwischen großen und kleinen Buchstaben. Verwenden Sie daher die Taste Caps-Lock bzw. Shift-Lock (= Feststelltaste für die Großschreibung) mit Bedacht: Im allgemeinen ist es nicht sinnvoll, wenn Sie die Taste benutzen, da Sie die Grundwörter von Dr. Logo stets in kleinen Buchstaben eingeben müssen.

Lesen Sie jetzt weiter, und machen Sie sich mit der Tastatur vertraut, indem Sie Dr. Logo einige arithmetische Gleichungen zur Lösung auftragen. Nachfolgend die Regeln, die Sie im Moment dafür brauchen:

- o Drücken Sie die Return- bzw. Enter-Taste, wenn Sie wollen, daß Dr. Logo die von Ihnen angegebenen Anweisungen befolgen soll. Wenn nichts passiert, haben Sie vielleicht vergessen, Return zu drücken.
- o Wenn Sie Tippfehler machen, verwenden Sie einfach die Del-Tasten (delete = löschen) oder die C1r-Taste (= Rücktaste), um das Zeichen unter oder links vom Cursor zu löschen.
- o Im vorliegenden Handbuch erscheint das, was Sie eintippen, **fettgedruckt**. Beachten Sie, daß Sie das Fragezeichen (?) nicht eintippen müssen, da es automatisch erscheint.

### Übung 1-1: Dr. Logo als Rechner einsetzen

Ihre Angaben	Erläuterung
?5-1 4	+ bedeutet <i>plus</i> , - bedeutet <i>minus</i> , * bedeutet <i>mal</i> und / bedeutet <i>geteilt durch</i> . Es muß kein Leerzeichen eingegeben werden.
?2+2 4	
?(2+3)*5 25	Beachten Sie, daß Dr. Logo die Ausdrücke in Klammern zuerst berechnet.
?2+3*5 17	
?(2+4)*(1+6+6*2)/3 38	



### Dr. Logo's Leitfaden

Wenn Sie eine andere Antwort erhalten, sehen Sie sich nochmal an, was Sie geschrieben haben, und überprüfen Sie die Klammern. Sie können die Daten noch einmal eintippen oder mit der Tastenkombination Control-Y beim CPC6128 bzw. beim Joyce mit der Copy-Taste wieder auf den Bildschirm holen und mit den Del-Tasten bearbeiten.

Nachfolgend einige zusätzliche Richtlinien für den Dialog mit Dr. Logo:

- o Wenn Sie über den rechten Rand hinausschreiben, nimmt Dr. Logo das darüber hinausgehende Zeichen automatisch in die nächste Zeile. Ein Ausrufezeichen erscheint am Ende der Zeile. Beispiel:
 

?Dr. Logo druckt am Ende einer Zeile ei!  
n Ausrufezeichen, wenn er den von Ihnen!  
eingegebenen Text automatisch trennt.
- o Dr. Logo weiß nicht, wann ein Wort endet und ein anderes beginnt, außer wenn Sie ein Leerzeichen eingeben. Achten Sie auf die Leerzeichen, wenn Sie die Übungen durchführen.
- o Dr. Logo teilt Ihnen mit, wenn er eine Anweisung nicht versteht, indem er eine *Hilfe*-Meldung ausgibt. Nachfolgend ein Beispiel von zwei Dialogen mit Dr. Logo:

?pr 4 + 5

9

?pr4 + 5

I don't know how to pr4

pr ist eines der Kommandos, die Dr. Logo bereits versteht (d.h. ein Grundwort). Dagegen ist pr4 etwas, von dem Dr. Logo nicht weiß, was er damit tun soll.

- Sie können Dr. Logo jederzeit - egal, was er tut - anzeigen, daß er seine Tätigkeit abbrechen soll, indem Sie die Esc-Taste (oder auch die Control-Taste und gleichzeitig die mit dem Buchstaben G gekennzeichnete Taste, also Control-G) drücken oder beim Joyce die Stop-Taste betätigen. Dr. Logo gibt folgende Meldung aus:

```
Stopped!
?
```

Drücken Sie die Esc- bzw. die Stop-Taste immer dann, wenn Sie einen Vorgang abbrechen wollen und zum ?-Prompt zurückkehren wollen oder dann, wenn der Bildschirm leer wird.

## 2 Der Dialog mit Dr. Logo

Sie können Dr. Logo anweisen, Dinge zu tun, indem Sie Anweisungen an der Tastatur eintippen, die aussehen wie einfache deutsche Worte in der Befehlsform. In der folgenden Lektion sollen Sie einige dieser Kommandos kennenlernen und sehen, wie Sie sie eingeben müssen.

1. Sie sollen lernen, was ein Kommando ist.
2. Sie sollen Zeichnungen mit Hilfe der Igel-Grafik-Kommandos erstellen.

### 2.1 Was ist ein Kommando

Ein *Kommando* ist eine eingetippte Zeile, die Dr. Logo anweist, eine Tätigkeit auszuführen. Sie geben Kommandos ein, indem Sie eine Zeichenkette eintippen und die Return-Taste drücken. Tippen Sie jetzt einen beliebigen Text zur Übung ein. Es spielt keine Rolle, ob Dr. Logo das Wort kennt oder nicht.

```
?geschirrspülen
I don't know how to geschirrspülen
?springen
I don't know how to springen
```

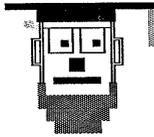
Mit diesem Handbuch werden Sie lernen, was Dr. Logo kennt und ausführen kann, und wie Sie ihm neue Tätigkeiten beibringen können. Während der Zeit, in der Sie lernen, können Sie praktisch alles eingeben - wie die obigen Beispiele zeigen - und brauchen sich keine Sorgen darüber zu machen, daß Sie Dr. Logo beschädigen könnten. Und wenn irgend etwas geschieht, das Sie nicht wollen, drücken Sie einfach auf die Control-Taste und gleichzeitig auf die Taste mit dem Buchstaben G bzw. beim Joyce die Stop-Taste. Dr. Logo bricht auf jeden Fall ab und gibt folgende Meldung aus:

```
Stoppend!
?
```

### 2.2 Zeichnungen mit den Igel-Grafik-Kommandos erstellen

Dr. Logo kennt verschiedene Grundwörter für das Bewegen des Igels. Wenn Sie diese verwenden, können Sie grafische Darstellungen entwerfen, die Linien und Kurven

enthalten. Das Ergebnis kann eine abstrakte Zeichnung, ein Bild oder eine Grafik sein.



### Dr. Logo's Leitfaden

Wenn Sie Dr. Logo starten, steht der Cursor links oben am Bildschirm. Er sollte für die Übungen jedoch zweigeteilt erscheinen: Der obere Teil für den Igel und die Zeichnung und die unteren fünf Zeilen für die Anzeige dessen, was Sie eingetippt haben. Wenn der Bildschirm also anders aussieht, dann drücken Sie die Control-Taste und gleichzeitig die Taste mit dem Buchstaben S (Control-S). Um den Bildschirm wieder ins Textformat zu ändern, drücken Sie Control-T (T steht für Text).

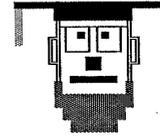
Versuchen Sie nun folgendes:

- o Tippen Sie fd ein
- o Drücken Sie die Leertaste
- o Tippen Sie 50 ein
- o Drücken Sie Return



```
?fd 50
?
```

fd (forward) bewegt den Igel um die Anzahl von Schritten weiter, die nach dem Kommando eingegeben werden. In diesem Fall geht der Igel 50 Schritte vorwärts. 50 wird *Eingabe* genannt, da diese Zahl in das Kommando *eingegeben* wird.



### Dr. Logo's Leitfaden

Wenn sich der Igel nicht vorwärts bewegt, überprüfen Sie Ihre Eingabe auf Tippfehler und eventuell fehlendes Leerzeichen.

- o Stellen Sie sicher, daß Sie nach Angabe des Kommandos die Return-Taste gedrückt haben.
- o Wenn Sie fd falsch geschrieben haben, meldet sich Dr. Logo etwa folgendermaßen:

```
I don't know how to df
```

- o Wenn Sie das Leerzeichen zwischen fd und der Eingabe von 50 vergessen haben, gibt Dr. Logo die folgende Meldung aus:

```
I don't know how to fd50
```

Tippen Sie das Kommando noch einmal ein.



```
?cs
?
```

cs (clear screen) löscht den Bildschirminhalt und hat zur Folge, daß der Igel in seine Ausgangsstellung *home*, d.h. auf die Mitte des Bildschirms, zurückkehrt.

ct (clear text) löscht den gesamten Text vom Bildschirm; das ?-Prompt wird wieder angezeigt und wartet auf das nächste Kommando.



?bk 60  
?

**bk** (back) bewegt den Igel um die Anzahl von Schritten rückwärts, die Sie als Eingabe angegeben haben. In diesem Fall ist die Eingabe 60.



?rt 90  
?

**rt** (right) dreht den Igel soviel Grad nach rechts, wie Sie als Eingabe angegeben haben. Im vorliegenden Fall ist die Eingabe 90.



?fd 15  
?lt 45  
?

**lt** (left) dreht den Igel um soviel Grad nach links, wie Sie als Eingabe angegeben haben. Im vorliegenden Fall ist die Eingabe 45.

Löschen Sie den Bildschirm durch Eingabe von `cs`, und drücken Sie Return, ehe Sie weitere Grundwörter ausprobieren.



?cs  
?pu  
?fd 30  
?

**pu** (pen up) bewirkt, daß der Igel keinen Strich zeichnet. Es ist keine Eingabe nötig.



?pd  
?fd 30  
?

**pd** (pen down) bewirkt, daß der Igel wieder eine Linie zeichnet. Auch hier ist keine Eingabe erforderlich.



?ht  
?

**ht** (hide turtle) läßt den Igel vom Bildschirm verschwinden. Wenn Sie ein Bild gezeichnet haben, können Sie **ht** eintippen, so daß Sie den gesamten Entwurf ansehen können.



?st  
?

**st** Mit **st** (show turtle) erscheint der Igel wieder.

Befolgen Sie die Schritte im nächsten Beispiel, um mit Dr. Logo ein Quadrat zu zeichnen.

### Übung 2-1: Figuren mit dem Igel zeichnen

Ihre Angaben

Reaktion des Igels

?cs 

?fd 50 

?rt 90 

?fd 50 

?rt 90 

?fd 50 

?rt 90 

?fd 50 

?rt 90

Der Igel zeigt jetzt in die gleiche Richtung wie beim Start, und Sie haben ein Quadrat gezeichnet!

?cs

Löschen Sie damit den Bildschirminhalt.

Es gibt auch einen einfacheren Weg, um ein Quadrat zu zeichnen.

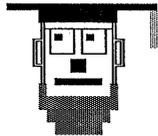
Beachten Sie, daß die Anweisungen **fd 50** und **rt 90** viermal wiederholt wurden. Sie können diese beiden Anweisungen in eine Liste eingeben, um nur ein Kommando anstelle von acht schreiben zu müssen. Dafür brauchen Sie ein neues Grundwort: **repeat**.

**repeat** benötigt zwei Eingaben: Eine Zahl und eine Liste von Anweisungen in Klammern. Es weist Dr. Logo an, die Liste mit den Anweisungen so oft zu wiederholen, wie die Zahl angibt.

Versuchen Sie es einmal:



```
?cs
?repeat 4 [fd 50 rt 90]
?
```



### Dr. Logo's Leitfaden

- o Wenn Dr. Logo angezeigt hat:  
I don't know how to fd50  
haben Sie zwischen Grundwort und Eingabe ein Leerzeichen vergessen.
- o Wenn Dr. Logo etwa folgende ähnliche Meldung ausgegeben hat:  
I don't know how to repat  
haben Sie repeat falsch eingetippt.
- o Wenn Dr. Logo reagiert hat:  
Not enough inputs to repeat  
haben Sie die eckigen Klammern vergessen oder etwas anderes eingetippt (wie z.B. runde Klammern).

Machen Sie weiter, und tippen Sie die Zeile noch einmal ein.

Sie können das Grundwort `repeat` auch mehr als nur einmal in einem Kommando verwenden - zumindest, wenn jedesmal danach eine Ziffer und eine Liste mit Anweisungen in eckigen Klammern steht. (Beim Joyce entspricht die eckige Klammer auf (I) dem Ä und die eckige Klammer zu (I) dem Ü.) Versuchen Sie einmal folgendes:



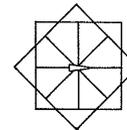
```
?cs
?repeat 4 [rt 90 repeat 4 [fd 50 rt 90 ]]
?
```

Sehen Sie sich die Eingaben in diesem Beispiel an. Es ist leichter, die Zeilen von rechts nach links zu lesen, anstatt wie üblich von links nach rechts.

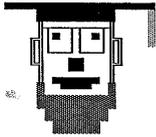
```
repeat 4 [rt 90 repeat 4 [fd 50 rt 90]]
```

The diagram illustrates the parsing of the command `repeat 4 [rt 90 repeat 4 [fd 50 rt 90]]`. It shows two levels of nesting. The outer level has a 'Zahl' (4) and a 'Liste mit Anweisungen' (the rest of the command). The inner level also has a 'Zahl' (4) and a 'Liste mit Anweisungen' (the inner command). Arrows point from the labels to the corresponding parts of the command.

Noch besser, versuchen Sie das Folgende:



```
?cs
?repeat 2 [rt 45 repeat 4 [rt 90 repeat!
4 [fd 50 rt 90]]]
```



## Dr. Logo's Leitfaden

Wenn Dr. Logo einen Teil einer Anweisung nicht versteht, gibt er eine *Hilfe*-Meldung aus. Wenn Sie eine Meldung ähnlich der folgenden sehen:

I don't know what to do with ...

Überprüfen Sie die Eingabe darauf, ob Sie die eckigen Klammern überall eingetippt haben, wo sie notwendig sind. Denken Sie daran, daß Sie immer eckige Klammern nach der nach repeat stehenden Zahl benötigen.

Wenn Sie die folgende Meldung sehen:

I don't know how to ...

haben Sie das Leerzeichen zwischen dem Grundwort (repeat, fd oder rt) und dessen Eingabe vergessen. Gehen Sie noch einmal in die Zeile, und benutzen Sie die Del- oder Clr-Taste, um den Tippfehler zu korrigieren.

### Übung 2-2: Verwendung von Listen für den Igel

Ihre Angaben

Erläuterung

?repeat 4 [fd 50 rt 90]



Der Igel wiederholt die Liste mit zwei Anweisungen viermal und zeichnet so ein Quadrat.

?repeat 3 [fd 50 rt 120]



Der Igel wiederholt die Liste mit Anweisungen dreimal und zeichnet so ein Dreieck. Der Winkel beträgt 120 statt 60 Grad, weil der Igel außerhalb des Dreiecks und nicht innerhalb gedreht werden muß.

120 Grad



?repeat 360 [fd 1 rt 1]



Der Igel wiederholt die Liste mit zwei Anweisungen 360mal und zeichnet so einen Kreis.

## 2.3 Zusammenfassung

Was Sie Neues gelernt haben:

- Sie können Dr. Logo anweisen, bestimmte Tätigkeiten durchzuführen, indem Sie ein Kommando eintippen und die Return-Taste drücken.
- Einige Kommandos benötigen eine Eingabe, damit sie vollständig sind.

Neue Grundwörter:

Name	Eingabe
fd	1 Zahl
bk	1 Zahl
rt	1 Zahl
lt	1 Zahl
pu	keine
pd	keine
cs	keine
ct	keine
ht	keine
st	keine
repeat	2 (Zahl, Liste mit Anweisungen)

### 3 Neue Prozeduren einfügen

Ein großer Fortschritt bei der Verwendung von Dr. Logo ist es, ihn dazu zu bringen, daß er sich an eine Reihe von Anweisungen wieder erinnert. Das geschieht, indem man eine Prozedur benennt und definiert und den Prozedurnamen in den Wortschatz von Dr. Logo mit aufnimmt. In der folgenden Lektion sollen Sie lernen, wie Prozeduren geschrieben werden.

#### Lernziele

1. Sie sollen eine Prozedur mit Igel-Grafik-Kommandos schreiben.
2. Sie sollen lernen, wie man Namen für Prozeduren wählt.

#### 3.1 Eine Prozedur schreiben

Um Dr. Logo ein neues Wort für eine Prozedur beizubringen, halten Sie sich an folgende Schritte:

- o Tippen Sie das Wort `to`, drücken Sie die Leertaste, und tippen Sie dann das Wort ein, das Sie definieren wollen.
- o Tippen Sie die Anweisungen. Tippen Sie eine Anweisung pro Zeile, und drücken Sie die Return-Taste, um in die nächste Zeile zu gelangen.
- o Um die Definition der Prozedur abzuschließen, muß man `end` eintippen. Dr. Logo gibt folgende Meldung aus:

Word defined

Nachfolgend ein Beispiel:

```
?to quadrat
>repeat 4 [fd 30 rt 90]
>end
quadrat defined
?quadrat
?
```



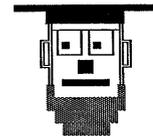
Wann immer Sie jetzt `quadrat` eintippen, bewegt Dr. Logo den Igel so, daß ein Quadrat gezeichnet wird. Versuchen Sie es. Der Prozedurname `quadrat` ist ein neues Wort im Vokabular von Dr. Logo.

Die nächste Übung zeigt, wie Sie eine Prozedur zum Zeichnen eines Fünfecks schreiben.

**Anmerkung:** Wenn Sie eine Prozedur mit mehr als fünf Zeilen schreiben, können Sie Control-T bzw. Alt-T drücken, damit der gesamte Bildschirm für die Texteingabe zur Verfügung steht. Der Bildschirm wird automatisch aufgeteilt, um für den Igel Platz zu machen, wenn Sie eine Igel-Grafikprozedur ausführen.

### Übung 3-1: Eine Prozedur schreiben

Ihre Angaben	Erläuterung
?360/5 72	Rechnen Sie erst aus, welche Winkel Sie benötigen. Dafür müssen Sie 360 durch die Anzahl der Seiten teilen. Sie können Dr. Logo als Rechner benutzen.  Die Winkel betragen jeweils 72 Grad. Die Seiten können beliebig lang sein. Verwenden Sie in dieser Prozedur 40 als Eingabe zur Vorwärtsbewegung.
?to fuenf	Wenn Sie to und danach ein Wort eintippen und die Return-Taste drücken, wird das ?-Prompt durch das Symbol > ersetzt. Dies bedeutet, daß Dr. Logo Ihre Anweisungen erst einmal registriert und sie nicht sofort ausführt.
>repeat 5 [fd 40 rt 72]	Dies ist die Definition eines neuen Wortes im Wortschatz von Dr. Logo oder die Anweisungen für die Prozedur.
>end fuenf defined ?	Dies bedeutet, daß Dr. Logo die Definition von fuenf, die Sie vorgenommen haben, jetzt kennt.
?fuenf	Probieren Sie jetzt die Prozedur aus.



### Dr. Logo's Leitfaden

- o Wenn Ihre Zeichnung nicht so wie die oben dargestellte aussieht, überprüfen Sie die Anweisungen auf dem Bildschirm, und vergleichen Sie sie mit den nachfolgenden:

```
to fuenf
repeat 5 [fd 40 rt 72]
end
```

Wenn Sie einen Fehler feststellen, tippen Sie **cs** ein, und drücken Sie die Return-Taste. Schreiben Sie die Prozedur noch einmal, und verwenden Sie einen anderen Namen, wie z.B. fuenfa. Dr. Logo erinnert sich dann an die Anweisungen für fuenf und fuenfa.

- o Wenn Dr. Logo wie folgt reagiert:

```
I don't know how to fuenf
```

haben Sie etwas anderes als fuenf als Prozedurnamen geschrieben. Verwenden Sie das Grundwort **pops**, um Ihre Prozeduren aufzulisten. So können Sie nachsehen, wie Sie Ihre Prozedur genannt haben. Geben Sie dieses Wort anstelle von fuenf ein.

Sie können fuenf als Kommando entweder alleine oder als Teil einer anderen Anweisung verwenden.

Jetzt können Sie Familie so editieren, daß r1 nicht erscheint:

```
?to Familie
>pr [Wie heisst du?]
>make "wahl r1
>if :wahl = [Carola]
> [pr [Guten Tag, Mami] stop]
>if :wahl = [Werner]
> [pr [Guten Tag, Papi] stop]
>if :wahl = [Bianca]
> [pr [Guten Tag, kleine Schwester] stop]
>end
Familie defined
```

Testen Sie jetzt die Prozedur.

```
?Familie
Wie heisst du?
Bianca
Guten Tag, kleine Schwester
?
```

## 10.4 Zusammenfassung

Neue Grundwörter:

Name	Eingaben
lput	2 (ein Objekt und eine Liste)
fput	2 (ein Objekt und eine Liste)
se,	2 Listen
shuffle	1 Liste
rl	keine

Neue, von Ihnen definierte Prozeduren:

Name	Eingaben
spass	2 Wörter
rezept	keine
weltraum	eine
weltall	keine
weltall1	keine
weltall2	keine
Familie	keine

## 11 Fehler suchen und beseitigen

Wenn Sie weiter mit Dr. Logo programmieren und Ihre Programmierkenntnisse erweitern, passiert es sicherlich manchmal, daß Sie eine Prozedur schreiben, die fehlerlos aussieht. Sie arbeitet jedoch nicht so, wie Sie geplant haben, bzw. sie arbeitet überhaupt nicht. Die Prozedur enthält einen Fehler. Vielleicht haben Sie schon jemand sagen hören: *Das Programm läuft großartig, wenn ich erst einmal die Fehler herausgefunden habe!* In dieser letzten Lektion sollen Sie lernen, wie Sie in Ihren Dr.-Logo-Programmen Fehler beseitigen bzw. Fehler finden.

### Lernziele

1. Sie sollen lernen, jede Zeile einer Prozedur einzeln zu betrachten.
2. Sie sollen lernen, Fehler zu beseitigen.

### 11.1 Beobachten Sie Prozeduren bei der Ausführung

Sie können Fehler in den Dr.-Logo-Prozeduren dadurch finden, daß Sie Ihre Prozedur bei der zeilenweisen Ausführung beobachten. Dadurch können Sie sich der Fehlerquelle bis in eine einzelne Zeile hinein nähern. Dies können Sie mit Hilfe des Grundwortes watch tun.

**watch** benötigt keine Eingabe. Es weist Dr. Logo an, jede Zeile einer Prozedur und deren Ergebnis nach Ausführung anzuzeigen. watch macht vor jeder Zeile eine Pause, bis Sie die Return-Taste drücken.

```
?watch
```

```
?quadrat
```

```
[1] In quadrat, repeat 4 [fd 50 rt 90]
```

Die Zahl in den eckigen Klammern zeigt die Ebene der Prozedur dieser Zeile. Die Prozedur, die Sie ausführen, hat Ebene 1, die Prozeduren in deren Definition haben Ebene 2. Beispiel: In der folgenden Anzeige hat haus Ebene 1. Die Prozeduren dach und rahmen sind Teil der Definition von haus und daher Ebene 2. Falls die Definition von rahmen die Prozedur kästchen enthielte, wäre kästchen eine Prozedur der Ebene 3 usw.

```
?haus
```

```
[1] In haus, rahmen
```

```
[2] In rahmen, repeat 4 [fd 30 rt 90]
```

```
[1] In haus, fd 30
```

```
[1] In haus, dach
[2] In dach, rt 30 fd 30
[2] In dach, fd 30
[2] In dach, rt 120 fd 30
[2] In dach, rt 30
```

(Immer, wenn Sie Return drücken, zeichnet der Igel den entsprechenden Teil der Zeichnung auf.)

`nowatch` weist Dr. Logo an, `watch` auszuschalten.

**Anmerkung:** Vergessen Sie nicht, `nowatch` einzutippen, wenn Sie die Fehlersuche beendet haben! Es passiert zwar nichts, wenn Sie es nicht tun, die Ausführung Ihrer Prozeduren benötigt aber eine Menge Zeit, wenn Sie nach jeder Zeile Return drücken müssen.

## 11.2 Eine Prozedur schreiben und Fehler beseitigen

Zum praktischen Üben der Fehlersuche schreiben Sie eine Prozedur, die von einer bestimmten Zahl an abwärts zählt. Überprüfen Sie, ob Sie die Fehlerstelle finden können, bevor Sie `watch` verwenden.

### Übung 11-1: Eine Prozedur von Fehlern befreien

```
Ihre Angaben
?to abwärts :zahl
>if :zahl = 0 [stop]
>op :zahl
>abwärts :zahl - 1
>end
abwärts defined
```

#### Erläuterung

Die Prozedur `abwärts` arbeitet nicht wie geplant. Verwenden Sie `watch`, um den Fehler herauszufinden.

```
?abwärts 3
3
```

An dieser Stelle wurde die Prozedur beendet, ohne daß Sie das wollten.

```
?watch
?abwärts 3
[1] In abwärts, if :zahl = 0 !
[stop]
[1] In abwärts, op :zahl
3
?
```

Sehen Sie sich diese Zeile an, und überlegen Sie, was die Prozedur getan hat: Sie hat das letzte Element der Liste ausgegeben. Fragen Sie sich, warum. Die Antwort ist, daß `op` die Prozedur gestoppt hat; `abwärts` stoppt, sobald die Eingabe ausgegeben wird!

```
?ed "abwärts
to abwärts :zahl
if :zahl = 0 [stop]
op :zahl
abwärts :zahl - 1
end
```

Machen Sie weiter, und editieren Sie abwärts.

```
to abwärts :zahl
if :zahl = 0 [stop]
pr :zahl
abwärts :zahl - 1
end
```

Tippen Sie `pr` anstelle von `op` ein.

```
?abwärts 3
[1] In abwärts, if :zahl = 0 !
[stop]
[1] In abwärts, pr :zahl 3
[1] In abwärts, abwärts :zahl!
- 1
[2] In abwärts, if :zahl = 0 !
[stop]
[2], pr :zahl 2
[2] In abwärts, abwärts :zahl!
- 1
[2] In abwärts, if :zahl = 0 !
[stop]
[2] In abwärts, pr :zahl 1
```

Führen Sie `abwärts` jetzt noch einmal aus.

Dieses Mal ist die Ausgabe so, wie Sie sie haben wollen. Die Prozedur arbeitet.

Der praktische Umgang im Fehlerbeseitigen macht Sie beim Programmieren mit Dr. Logo immer sicherer. Wenn Sie mit einer Prozedur beginnen, die dem von Ihnen er-

wünschten Ergebnis sehr nahe kommt, können Sie so lange nach Fehlern suchen, bis sie einwandfrei läuft!

### 11.3 Zusammenfassung

Neue Grundwörter:

Name	Anzahl der Eingaben
watch	keine
nowatch	keine

Neue, von Ihnen definierte Prozeduren:

Name	Anzahl der Eingaben
abwärts	1 Liste

### Stichwortverzeichnis

- abbrechen, S-14
- ändern, S-33
- Anführungszeichen, S-40
- Anweisung, S-8, S-15
- Arbeitsdiskette, S-10
- Arbeitsspeicher, S-9, S-39
  - organisieren, S-58
  - überprüfen, S-39, S-42
- Ausgabe, S-51
- Ausrufezeichen, S-13
  
- Balkendiagramm, S-61
- Bedingung, erfüllen S-82
- beenden, S-59 \*
- benennen, S-70
- Betriebssystem, S-9
- bf, S-61
- Bildschirm, S-16
  - löschen, S-17
- bk, S-18
- bl, S-61
  
- Caps-Lock, S-11
- Clr, S-12, S-34
- Control-G, S-14
- Control-S, S-16
- Control-T, S-16
- Control-Y, S-13
- Copy, S-13
- cs, S-17
- ct, S-17
  
- Darstellung, grafische S-8
- Date, laden S-40
- Datei, S-39
  - löschen S-41
- Dateinamen, S-39
- Daten-Diskette, S-40
- Del, S-12, S-34
- Dialog, S-15
  
- Dialogverkehr, S-7
- Diskette, S-39
  - leere, S-40
- Doppelpunkt, S-45, S-70
- Dr. Logos Leitfaden, S-8
  
- Ebene, S-87
- ed, S-34
- editieren, S-33
- Editor, S-33
- Eingabe, S-16, S-45
- Eingaben, wiederholen S-57
- Element, ersetzen S-77
  - erstes, S-61
  - letztes, S-61
- Elemente, mischen S-80
- empty, S-63
- Enter, S-12
- er, S-42
- erall, S-42, S-43
- erasefile, S-41
- Esc, S-14
- experimentieren, S-8
  
- FALSE, S-63
- fd, S-16
- Fehler, beseitigen S-87
  - suchen, S-87
- Fehlersuche, S-88
- first, S-61
- formatieren, S-9, S-40
- fput, S-77
  
- gprop, S-74
- Grafik, zeichnen S-8
- Grafikprozedur, S-28
- Großschreibung, S-30
- Grundwort, S-7
- Grundwortnamen, S-31

Hilfe-Meldung, S-13  
 ht, S-20

if, S-59

Igel, bewegen S-15  
 \* erscheint, S-20  
 links, S-19  
 rechts, S-18  
 rückwärts, S-18  
 verschwindet, S-20

Kernbefehl, S-7  
 Kleinschreibung, S-30  
 Kommando, S-13, S-15  
 eingeben, S-16  
 Kopieren, S-11  
 Kurve, S-15

laden, S-40  
 last, S-61  
 Laufwerk, S-9  
 Linie, S-15  
 links, S-19  
 list, S-69  
 Liste, S-23, S-61  
 Listen, zusammenlegen, S-80  
 load, S-40  
 löschen, S-39  
 lput, S-77  
 lt, S-19

make, S-70  
 Merkmal, S-74  
 anzeigen, S-74  
 zuordnen, S-74  
 Merkmalliste, S-67, S-73

nowatch, S-88

Objekt, S-67  
 op, S-51  
 Originaldiskette, S-9

pd, S-19  
 Pfeiltasten, S-34  
 Platzhalter, S-45, S-70  
 plist, S-74  
 po, S-42  
 pons, S-72  
 pops, S-29  
 pots, S-42  
 pprop, S-74  
 Primitive, S-7  
 Programm, erstellen S-51  
 Programmausführung, beobachten, S-87  
 Prompt, S-10  
 Prozedur, S-7  
 ändern, S-33  
 beenden, S-59  
 benennen, S-27  
 editieren, S-33  
 einfügen, S-27, S-27  
 größere, S-53  
 löschen, S-42  
 schreiben, S-27, S-28  
 sichern, S-39  
 variable, S-45  
 Prozedurdefinition, ändern S-33  
 anzeigen, S-42  
 Prozedurebene, S-87  
 Prozedurname, S-27, S-30, S-33, S-67  
 pu, S-19

rechts, S-18  
 recycle, S-58  
 repeat, S-21, S-23  
 Return, S-12  
 rl, S-82  
 rt, S-18, S-18

Satz, anzeigen S-77  
 save, S-39, S-73  
 Schleife, unendliche S-57  
 schreibgeschützt, S-9  
 Shift-Lock, S-11  
 sichern, S-39

Sicherungskopie, S-10  
 st, S-20  
 starten, S-11  
 Stop, S-14, S-59  
 System-Prompt, S-9

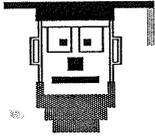
Tastatur, S-11  
 Text,löschen S-17  
 Text-Bildschirm, S-68  
 Texteditor, S-33  
 Texteingabe, S-28  
 Textformat, S-16  
 thing, S-70  
 to, S-27  
 TRUE, S-59, S-63

Umlaute, S-11

Vorgang, abbrechen S-14

watch, S-87  
 ausschalten, S-88  
 Wort, neues S-28  
 Wörter, S-67  
 eintippen S-67  
 Wörterliste, S-69  
 Wortspiel, S-8, S-77

Zeile, ausgeben S-82  
 Ende der S-13

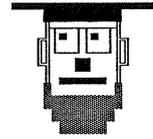


## Q

### quotient *nn*

Gibt den ganzzahligen Teil des Quotienten aus der ersten und zweiten Zahl aus. Im Gegensatz zum Operator / schneidet `quotient` bei den eingegebenen Zahlen den Teil hinter dem Komma vor Ausführung der Division ab.

```
?quotient 10 4
2
?/ 10 4
2.5
?10 / 4
2.5
?10 quotient 4
10
Not enough inputs to quotient
?quotient 5 .9
Can't divide by zero
```



## R

### random *n*

Gibt eine ganze, nicht negative Zufallszahl aus, die kleiner ist als die eingegebene Zahl. Weiteres über `random` finden Sie in der Erklärung von `rerandom`. Im folgenden Beispiel ist `char 9` ein Tabulator.

```
?random 10
4
?repeat 20 [(type random 10 char 9)]
3 0 2 3 1 4 6 4 4 7 !
9 4 5 1 8 2 6 8 9 9'!
```

### rc

Gibt das erste Zeichen aus, das über die Tastatur eingegeben wurde, einschließlich Kontrollzeichen Funktionstasten, außer `Esc` bzw. `Stop` und `Control-Z`. Der `rc`-Befehl bewegt weder den Cursor, noch druckt er das eingegebene Zeichen auf dem Textfenster aus. Mit `keyp` können Sie sehen, ob bereits ein Zeichen zum Lesen eingegeben wurde.

Wenn Sie `Control-Z` benutzen, während `rc` auf eine Eingabe wartet, müssen Sie `co` eingeben, um die Ausführung der Prozedur fortsetzen zu lassen. (Siehe `co`).

```
?rc
z
?
?to quiz
>type [Kennst du einen Marsmenschen (J\N)?\ ]
>if "J = uc rc
> [pr [Was, du glaubst daran?] stop]
```

```
> [pr [Ich auch nicht!]]
>quiz
>end
quiz defined
?quiz
Kennst du einen Marsmenschen (J/N)? n
Ich auch nicht!
Kennst du einen Marsmenschen (J/N)? j
Was, du glaubst daran
```

## recycle

Gibt so viele Knoten wie möglich frei und organisiert den Arbeitsspeicher um. recycle aktiviert das Säuberungsprogramm. In Kapitel 6 finden Sie mehr Information über das Säuberungsprogramm. Wenn Dr. Logo die Meldung "Zuwenig Hauptspeicher" und ein Ausrufezeichen (!) ausgibt, müssen Sie sofort recycle eingeben.

```
?nodes
989
?recycle
?nodes
2379
```

## release\* Kanal\_n

Der release-Befehl gibt den Tonkanal wieder frei, der durch den sound-Befehl besetzt wurde. Die Kanäle werden dabei folgendermaßen gekennzeichnet:

Kanal_n	freigegebener Kanal
0	keiner
1	a
2	b
3	a und b
4	c
5	a und c
6	b und c
7	a, b und c

```
?release 4
?
```

\* Nicht auf dem Joyce implementiert.

## remainder n n

Gibt den ganzen Teil des Restes aus, der bei Division der ersten durch die zweite Zahl verbleibt. In der Erklärung von pause finden Sie in der Prozedur nautilus ein Anwendungsbeispiel von remainder in einer Prozedur.

```
?remainder 7 3
1
```

## remprop Name Merkmal

Entfernt das eingegebene Merkmal aus der Merkmalliste des Namens. Sie müssen remprop benutzen, um nicht systemeigene Merkmale aus der Merkmalliste eines Objektes zu entfernen, die durch Anwendung des Kommandos pprop in die Merkmalliste eingetragen wurden. Um einen Namen aus dem Arbeitsspeicher vollständig löschen zu können, müssen alle Merkmale aus der Merkmalliste dieses Objektes gelöscht sein.

```
?pons
Pferd1 ist Attila
Pferd2 ist Blitz
?plist "Pferd1
[Farbe Rappe .APV Attila]
?ern "Pferd1
?plist "Pferd1
[Farbe Rappe]
?remprop "Pferd1 "Farbe
?plist "Pferd1
[]
?plist "Pferd2
[Farbe [Schimmel] Begabung [Jagd Springen] .APV Blitz]
?
```

```
?to merkmal.weg :Name
>;Beseitigt alle Merkmale vom eingegebenen Namen
>if empty plist :Name [stop]
>make "Merkmal first plist :Name
>run (se "remprop
> " :Name " :Merkmal)
>merkmal.weg :Name
>end
merkmal.weg defined
```

```
?merkmal.weg "Pferd2
?plist "Pferd2
[]
?
```

## repeat n Befehlsliste

Führt die eingegebene Befehlsliste n mal aus. Der eingegebene Wert muß eine positive, ganze Zahl sein. Wenn die Eingabe keine positive, ganze Zahl ist, schneidet repeat den Teil hinter dem Komma ab. In der Erklärung von px, in der Prozedur demo.us finden Sie ein Anwendungsbeispiel für verschachtelte repeat-Ausdrücke. Der Eingabebereich von repeat geht von 0 bis 32768.

```
?make "seiten 3
?repeat :seiten [fd 40 rt (360 / :seiten)]
```

## rerandom

Läßt den nachfolgenden random- oder shuffle-Ausdruck eine Zufallszahlenfolge reproduzieren. Der Kernbefehl rerandom speichert die nächste Zufallszahlenfolge ab, die durch random- oder shuffle-Ausdrücke erzeugt wird. Dann wird bei jeder folgenden Eingabe von rerandom und des abgespeicherten random- oder shuffle-Ausdruckes die gleiche Zahlenfolge ausgegeben, unabhängig davon, wie viele random- oder shuffle-Ausdrücke Sie zwischendurch ausgeführt haben.

```
?rerandom
?repeat 10 [(type random 10 char 9)]
2 3 7 5 3 2 0 4 2 6 !
?random 365
210
?rerandom
?repeat 10 [(type random 10 char 9)]
2 3 7 5 3 2 0 4 2 6 !
?random 180
103
?rerandom
?repeat 10 [(type random 10 char 9)]
2 3 7 5 3 2 0 4 2 6 !
?rerandom
?shuffle [Äpfel Bananen Korn Hand]
```

```
[Korn Bananen Hand Äpfel]
?shuffle [Frau schlau braun blau]
[blau schlau braun Frau]
?rerandom
?shuffle [Äpfel Bananen Korn Hand]
[Korn Bananen Hand Äpfel]
?
```

## rl

Gibt eine Liste aus, die die über die Tastatur eingegebene Zeile enthält. Der Befehl rl kann eine Zeile erst lesen, nachdem Sie die Return-Taste gedrückt haben. Verwenden Sie keyp, um zu sehen daß bereits Zeichen eingegeben wurden und gelesen werden können.

Wenn Sie Control-Z benutzen, während rl auf eine Eingabe wartet, müssen Sie co eingeben, um die Ausführung der Prozedur fortsetzen zu lassen (siehe co).

```
?rl
juchee hurra
[juchee hurra]
?repeat 5 rl
fd 40 rt 72
?
```

## round n

Rundet die eingegebene positive Zahl auf die nächste ganze Zahl auf oder ab und gibt das Ergebnis aus. Eine negative Zahl wird auf die nächste ganze Zahl abgerundet. Ein Bruchteil von 0.5 oder größer wird aufgerundet. Der Eingabebereich des round-Kommandos geht von -32768.5 bis +32767.5.

```
?round 3.333333
3
?int 3.5
3
?round 3.5
4
```

**rq**

Gibt ein Wort aus, das alle Zeichen enthält, die in einer Zeile über die Tastatur eingegeben wurden, und gibt dieses Wort auf dem Textfenster aus. Der rq-Befehl kann eine Zeile erst lesen, nachdem Sie die Return-Taste gedrückt haben. Verwenden Sie keyp, um zu sehen, ob bereits Zeichen eingegeben wurden und gelesen werden können.

Wenn Sie Control-Z verwenden, während rq auf eine Eingabe wartet, müssen Sie `co` eingeben, um die Ausführung der Prozedur fortsetzen zu lassen (siehe `co`).

```
?rq
```

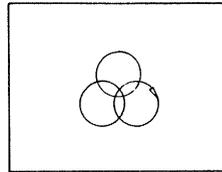
```
Ich glaube nicht, dass wir noch in Bayern sind
```

```
Ich glaube nicht, dass wir noch in Bayern sind
```

**rt** Grad\_n

Dreht den Igel um den in Grad eingegebenen Winkel nach rechts (im Uhrzeigersinn). Eine negative Zahl dreht den Igel nach links. Sie können jede Zahl von -32767 bis +32767 eingeben.

```
?to gehe.rechts :radius :anzahl
>seth 270 ht
>repeat :anzahl
> [rt 360 / :anzahl
>  kreis.rechts :radius]
>end
gehe.rechts defined
?to kreis.rechts :radius
>repeat 60
> [fd (:radius *
>  (2.618 * 6 / 5) / 30)
>  rt 6]
>end
kreis.rechts defined
?gehe.rechts 50 3
```

**run** Befehlsliste

Führt die eingegebene Befehlsliste aus. Wenn die eingegebene Befehlsliste ein Objekt ausgibt, gibt run dieses Objekt aus. Sie können `list` oder `se` verwenden, um eine

Befehlsliste für run zusammensetzen. In der Erklärung von `remprop` finden Sie ein Anwendungsbeispiel von run in einer Prozedur.

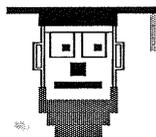
```
?run [pr [grün]]
```

```
grün
```

```
?run [empty 0 9]
```

```
FALSE
```

```
?run (se [cs] [setpc 3][repeat 4 [fd 40 rt 90]])
```



## S

### save Datname

Schreibt den Inhalt des Arbeitsspeichers oder der angegebenen Pakete in die angegebene Disketten-Datei und fügt zum Namen dieser Datei den Dateityp `.LOG` zu. Von `save` werden alle Prozedur- und Variablendefinitionen und alle Merkmalszuordnungen, die sich im Arbeitsspeicher befinden, in die Datei kopiert. Wenn Sie eine andere Kommandozeile (außer einem `pause`-Ausdruck) in eine Datei abspeichern wollen, müssen Sie das Kommando `edf` benutzen.

Die Länge des Dateinamens darf acht Zeichen nicht überschreiten. Wenn der Dateiname bereits vergeben ist, müssen Sie entweder einen anderen Namen wählen, die alte Datei mit `erasefile` löschen oder mit `change` umbenennen, bevor Sie die Datei abspeichern.

```
?save "ursula
?save "insekten
?save "figuren
?dir
[FIGUREN INSEKTEN URSULA]
```

### savepic Datname

Schreibt den Inhalt der Zeichenfläche in die bei der Eingabe genannte Bilddatei und fügt zum Namen dieser Datei den Dateityp `.PIC` hinzu.

Die Länge des Dateinamens darf acht Zeichen nicht überschreiten. Wenn der Dateiname bereits vergeben ist, müssen Sie entweder einen anderen Namen wählen, die alte Datei mit `erasepic` löschen oder mit `change` umbenennen, bevor Sie die Datei abspeichern.

```
?dirpic
[BLUME GESICHT]
?to drehe.quadrat
>repeat 12 [repeat 4 [fd 50 rt 90]
>   rt 30]
>end
```

```
drehe.quadrat defined
?drehe.quadrat
?savepic "quadrate
?dirpic
[QUADRATE BLUME GESICHT]
```

### se Objekt Objekt (...)

Gibt eine Liste aus, die aus allen eingegebenen Objekten ohne die äußeren Klammern zusammengesetzt ist. Vergleichen Sie mit `list`. Sie können `list` oder `se` verwenden, um eine Eingabeliste für die meisten Prozeduren (außer `if`) zu erstellen, die *literale Listen* (Listen in eckigen Klammern) als Eingabe benötigen.

```
?make "Igel (se rl rl rl)
Vanille
Karamell
Schokolade
?:Igel
[Vanille Karamell Schokolade]
?se "Katze [Hase Kaninchen]
[Katze Hase Kaninchen]
?se [langsam und ausdauernd] [gewinnt man das Rennen.]
[langsam und ausdauernd gewinnt man das Rennen.]
?
```

### setbg\* Farbe\_n

Färbt den Hintergrund der Zeichenfläche und des Textfensters von `ss` der eingegebenen Farbnummer entsprechend ein. Die größte Farbnummer beim CPC6128 ist 3 (und 0 beim Joyce).

\*Nur sinnvoll auf dem CPC6128

```
?to sehe.hgfarben :farbe
>if :farbe < 0 [stop]
>setbg :farbe
>(pr [Dies ist die Farbnummer] item 1 sf)
>sehe.hgfarben :farbe - 1
>end
sehe.hgfarben defined
?sehe.hgfarben 3
```

### setcursor Koord\_Liste

Setzt den Cursor auf die durch die eingegebenen Koordinaten bestimmte Textfensterposition. Das erste eingegebene Element ist die Spaltennummer oder X-Koordinate; das zweite Element ist die Zeilennummer oder Y-Koordinate.

```
?to hallo
>ct
>repeat 20
> [setcursor list random 40 random 25
>   pr "hallo]
>pr "bye
>end
hallo defined
?hallo
bye
```

### setd d:

Macht das angegebene Laufwerk zum vorgegebenen Laufwerk. Bei Dateinamen ohne Laufwerksbezeichnung nimmt Dr. Logo an, daß sich die Datei auf der Diskette im vorgegebenen Laufwerk befindet.

```
?defaultd
A:
?dir
[WERKZEUG BEGINNE]
?setd "b:
?defaultd
B:
?
```

### seth Grad\_n

Dreht den Igel in die Richtung, die durch die eingegebene Anzahl von Winkelgraden angegeben wird. Unabhängig von seiner aktuellen Richtung dreht einen positive Zahl den Igel um die Anzahl von Graden von der Richtung Norden beginnend im Uhrzeigersinn (nach rechts). Eine negative Zahl dreht den Igel gegen den Uhrzeigersinn (nach links). Die Zahl muß im Bereich von -32767 bis +32767 liegen.

```
?seth 90
?seth 180
?seth -90
```

```
?to zeichnekompass :nummer
>if :nummer > 330 [seth 0 stop]
>pu seth :nummer
>fd 50
>bk 50 rt 23
>fd 45 pd
>fd 5 pu
>bk 50
>zeichnekompass :nummer + 45
>end
zeichnekompass defined
?zeichnekompass 0
?
```

### setpal\* Farb\_n RGB\_Liste

Der Kernbefehl setpal (set palette) setzt mit der RGB-Liste die Farbe für die angegebene Farbnummer fest. Die Farbnummer kann beim CPC6128 zwischen 0 und 3 liegen. Die Farbwerte für rot grün und blau können zwischen 0 und 2 variieren, wobei die Schrift bzw. der Zeichenstift am Monochrom-Bildschirm um so heller erscheinen, je größer die Zahl ist. Mit pal lassen Sie sich die RGB-Liste anzeigen.

```
?pal 1
[2 2 2]
?setpal 1 [1 2 1]
?pal 1
[1 2 1]
```

\*Nicht auf dem Joyce implementiert.

**setpc** Farbe\_n

Stellt die Farbe des Farbstiftes entsprechend der eingegebenen Farbnummer ein. Die eingegebene Zahl darf beim CPC6128 zwischen 0 und 3 liegen bzw. beim Joyce die Werte 0 und 1 haben. In Abschnitt 5.5.3 finden Sie weitere Information über die Farben der graphischen Zeichenfläche.

```
?setpc 1 fd 30
?setpc 0 fd 30
?setpc 1 fd 30
?setpc 0 fd 30
```

**setpos** Koord\_Liste

Verschiebt den Igel in die von der Eingabeliste festgelegte Position. Das erste Element ist die X-Koordinate und die zweite Zahl ist die Y-Koordinate. In Abschnitt 5.5.1 finden Sie weitere Information über das Koordinatensystem der Zeichenfläche.

```
?cs
?setpos [80 50]
?setpos [0 30]
?setpos [80 -50]
?setpos [0 0]
```

**setscrunch** n

Stellt das Bildseitenverhältnis entsprechend der eingegebenen Zahl ein. Das ermöglicht Verzerrungen und die Anpassung an Monitore mit anderen Bildpunktverhältnissen.

```
?to kreis
>repeat 360 [fd 1 rt 1]
>end
kreis defined
?kreis
?setscrunch 2
?kreis
?
```

**setsplit** n

Legt die Anzahl der Zeilen im Textfenster des geteilten Bildschirms fest. Die Zahl muß zwischen 1 und 25 liegen sollte.

```
?ts
?to drehe.quadrat
>repeat 12
> [repeat 4
> [fd 50 rt 90]
> rt 30]
>end
drehe.quadrat defined
?fs
?drehe.quadrat
?ss
?setsplit 3
?ts
?fs
```

**setx** n

Verschiebt den Igel auf die eingegebene X-Koordinate. Die Y-Koordinate wird nicht verändert. Wenn window gesetzt ist, können Sie auch setx-Werte außerhalb des sichtbaren Bereiches angeben, um den Igel außerhalb der Zeichenfläche zeichnen zu lassen. Wenn wrap gesetzt ist, verwandelt setx alle außerhalb des sichtbaren Bereiches liegenden, eingegebenen Koordinaten so, daß der Igel springt und sichtbar bleibt. In Abschnitt 5.5.1 finden Sie weitere Information über die Koordinaten der Zeichenfläche.

```
?setx -50
?setx 80
```

**sety** n

Verschiebt den Igel auf die eingegebene Y-Koordinate. Die X-Koordinate wird nicht verändert. Wenn window gesetzt ist, können Sie auch sety-Werte außerhalb des sichtbaren Bereiches angeben, um den Igel außerhalb der Zeichenfläche zeichnen zu lassen. Wenn wrap gesetzt ist, verwandelt sety alle außerhalb des sichtbaren Bereiches liegenden eingegebenen Koordinaten so, daß der Igel springt und sichtbar bleibt.

In Abschnitt 5.5.1 finden weitere Information über die Koordinaten der Zeichenfläche.

```
?sety 90
?sety -50
```

## sf

Gibt eine Liste aus, die die aktuellen Eigenschaften der Zeichenfläche enthält. Die Liste enthält fünf Elemente in folgender Anordnung:

```
[HGFARBE BILDMODUS TEXTZEILEN BERANDUNG PROP]
```

HGFARBE	(bgcolor_n) ist die aktuelle Nummer der Hintergrundfarbe.
BILDMODUS	(screen-state) gibt SS für Teilbildschirm-Modus, TS für Textbildschirm-Modus und FS für Vollgraphikbildschirm-Modus an.
TEXTZEILEN	(split-size) gibt die Anzahl der Textzeilen an, die auf der Zeichenfläche ausgegeben werden.
BERANDUNG	(window-state) gibt WINDOW, WRAP oder FENCE an.
PROP	(scrunch) gibt das aktuelle Bildseitenverhältnis der Zeichenfläche an.

In Kapitel 5 finden Sie ausführliche Information über verschiedene Aspekte der graphischen Zeichenfläche.

```
?setbg 0
?ts
?setscrunch 2
?sf
[0 TS 10 WINDOW 2]
```

## show Objekt

Gibt das eingegebene Objekt gefolgt von einem Wagenrücklauf auf dem Textfenster aus. show entfernt die äußeren Klammern einer Liste nicht. Sie müssen den Kernbefehl show nicht in der Hauptebene benutzen, er kann allerdings in einer Prozedur nützlich sein. Versuchen Sie die unten angegebenen Beispiele mit und ohne show-Kommando durchzuspielen und betrachten Sie das Ergebnis. Vergleichen Sie show mit pr und type.

```
?show "hallo
hallo
```

```
?show 5 + 4
9
?show [Krabbe Muschel]
[Krabbe Muschel]
?to addiere :n :m
>show + :n :m
>end
addiere defined
?addiere 2 3
5
```

## shuffle Liste

Gibt eine Liste bestehend aus Elementen der eingegebenen Liste in zufälliger Reihenfolge aus. In der Erklärung von rerandom finden Sie weitere Information über shuffle.

```
?shuffle [Fischt Fischers Fritz]
[Fischt Fischers Fritz]
```

## sin Grad\_n

Gibt den Sinus des in Grad angegebenen Winkels in Form einer Dezimalzahl zwischen 0 und 1 aus.

```
?sin 90
1
```

## sound\* Kanalstat Periode Dauer Lautst Lautst\_Hüllk Ton\_Hüllk Rauschen ✓

Der sound-Befehl von Dr. Logo nutzt die vielfältigen Möglichkeiten der Tonerzeugung beim CPC6128. Zusammen mit diesem Befehl können bis zu sieben Parameter übergeben werden. Sie müssen mindestens die beiden ersten angeben, den Kanalstatus und die Tonperiode, um einen Ton mit der Länge von 0,2 Sekunden zu erzeugen.

\* Nicht auf dem Joyce implementiert.



```
?repeat 10 [rt 36 fuenf]
?
```

**Anmerkung:** Immer, wenn Sie einen vollständigen Kreis zeichnen wollen, muß die Anzahl der Wiederholungen multipliziert mit der Richtungszahl 360 ergeben (d.h. die Anzahl von Graden in einem Kreis). Sie können Dr. Logo als Rechner für das Errechnen dieser Zahlen verwenden.

### 3.2 Namen für Prozeduren wählen

Nachfolgend einige Punkte, an die Sie sich erinnern sollten, wenn Sie einen Namen für eine Prozedur vergeben:

- o Damit Sie sich gut daran erinnern, wählen Sie einen Prozedurnamen, der die Tätigkeit der Prozedur beschreibt. Beispiele für Prozedurnamen, die Sie verwenden könnten, sind *quadrat* (um damit ein Quadrat zu zeichnen), *telbuch* (um Telefonnummern aufzulisten) oder *spiel* (um mit einem Wortspiel, das Sie erstellt haben, zu spielen).
- o Ein Prozedurname darf keine Leerzeichen enthalten; aber er kann Punkte enthalten. Sie können eine Prozedur *freund.tel* nennen, jedoch nicht *freund tel*. Einfach deshalb, weil Dr. Logo denkt, daß ein Wort zu Ende ist, sobald ein Leerzeichen auftritt.
- o Wenn Sie zwei ähnliche Prozeduren schreiben, denken Sie daran, zwei ähnliche Namen dafür auszuwählen (z.B. *quadr20*, *quadr70*).
- o Da Dr. Logo zwischen Groß- und Kleinschreibung unterscheidet, werden mit *DREIECKE*, *DreiEcke*, *Dreiecke* und *dreiecke* vier verschiedene Prozeduren benannt.

- o Da Dr. Logo die Prozeduren für die Grundwörter bereits kennt, können Sie keine Grundwortnamen für Ihre Prozeduren verwenden. Wenn Sie dies dennoch tun sollten, reagiert Dr. Logo mit folgender Anzeige:

```
Wort is a primitive
```

Wenn dies auftritt, wählen Sie einen anderen Namen.

### 3.3 Zusammenfassung

Was Sie Neues gelernt haben:

- o Sie können dafür sorgen, daß Dr. Logo sich an eine Reihe von Anweisungen erinnert, indem Sie eine Prozedur schreiben.
- o Sie schreiben Prozeduren dadurch, indem Sie *to*, dann den Namen der Prozedur, die Anweisungen und schließlich *end* eintippen.
- o Wann immer Sie den Prozedurnamen angeben, führt Dr. Logo die Anweisungen aus.
- o Sie können Prozedurnamen alleine oder als Teil einer anderen Anweisung verwenden.

Neue, von Ihnen definierte Prozeduren:

Name	Eingabe
quadrat	keine
fuenf	keine

## 4 Prozeduren ändern

Sie werden sehr oft eine Prozedurdefinition ändern müssen, nachdem Sie sie geschrieben haben. Vielleicht stellen Sie fest, daß ein Fehler in der Prozedur ist, da sie nicht arbeitet. Oder Sie wollen ganz einfach, daß die Prozedur etwas anderes ausführen soll. In dieser Lektion sollen Sie lernen, wie Sie Prozedurdefinitionen mit einem Hilfsmittel von Dr. Logo, dem Editor, ändern können.

### Lernziele

1. Sie sollen lernen, warum man eine Prozedurdefinition ändern können sollte.
2. Sie sollen lernen, wie der Editor einzusetzen ist.
3. Sie sollen die Prozedur fuenf editieren.

### 4.1 Warum man eine Prozedurdefinition ändern können sollte

Der Editor ist ein Hilfsmittel, das Sie zur Änderung der Definition von Prozeduren verwenden können. Es gibt zwei Möglichkeiten, bei denen Sie den Editor verwenden können:

1. Dr. Logo teilt Ihnen mit, daß er die Prozedur nicht vervollständigen kann. Beispielsweise gibt er folgende Meldung aus:

```
I don't know how to xyz  
Not enough inputs to xyz
```

Wenn Sie diese Hilfe-Meldungen sehen, müssen Sie die Prozedur editieren, um dem Problem auf den Grund zu kommen.

2. Sie wollen eine Prozedur so ändern, daß sie anders arbeitet. Beispielsweise wollen Sie eine Prozedur, die ein Quadrat von 40 Schritten zeichnet, so ändern, daß sie ein Quadrat von 60 Schritten zeichnet.

### 4.2 Wie der Texteditor zu verwenden ist

Es gibt zwei Möglichkeiten, den Texteditor einzusetzen:

1. Man tippt das Grundwort ed (edit) ein, drückt die Leertaste einmal, tippt ", tippt den Namen der Prozedur, die man ändern will und drückt dann auf die Return-Taste. Zwischen " und den Prozedurnamen darf kein Leerzeichen sein.

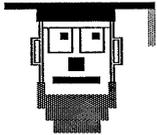
Wenn Sie das folgende Kommando eingeben:

```
ed "quadrat
```

erscheint folgendes auf Ihrem Bildschirm:

```
*.
to quadrat
repeat 4 [fd 50 rt 90]
end
```

- Wenn Sie eine Hilfe-Meldung sehen, tippen Sie das Grundwort `ed` ein und drücken auf die Return-Taste. Der Editor löscht den Bildschirminhalt und zeigt die Definition der Prozedur an, die Sie ausführen wollen. Der Cursor erscheint in der Zeile mit dem Fehler.



### Dr. Logo's Leitfaden

Wenn Sie `ed` eintippen und der Bildschirminhalt dann gelöscht wird, tippen Sie Alt- bzw. Control-G, um zurück zum `?`-Prompt zu kommen. Tippen Sie dann wieder `ed` und danach den Prozedurnamen ein.

Wenn Sie die Return-Taste drücken, erscheint die Definition auf dem ungeteilten Bildschirm (ohne Igelmarke), und Sie können mit Hilfe der folgenden Schritte editieren:

- Bewegen Sie den Cursor mit den Pfeiltasten an die Stelle, an der Sie Zeichen hinzufügen oder löschen wollen. (Weitere Einzelheiten über den Cursor finden Sie im Referenzhandbuch für Ihren Computer.)
- Verwenden Sie die `Del`- bzw. `+Del`-Taste, um das Zeichen links vom Cursor zu löschen, oder drücken Sie `Clr` bzw. `Del+` (oder Alt- bzw. Control-D), um das Zeichen direkt über dem Cursor zu löschen. Tippen Sie einen beliebigen, neuen Text ein, der Editor wird ihn automatisch einfügen. Wenn Sie eine ganze neue Zeile einfügen wollen, drücken Sie am Ende der vorangehenden Zeile auf die Return-Taste.

- Wenn Sie mit der neuen Definition zufrieden sind, drücken Sie Alt- bzw. Control-C oder beim Joyce die Exit-Taste. Dadurch wird der Editor angewiesen, die Definition der Prozedur entsprechend den von Ihnen gemachten Angaben zu ändern.
- Wenn Sie die Änderungen nicht sichern wollen, drücken Sie Alt- bzw. Control-G oder beim Joyce die Stop-Taste. Dadurch wird der Editor angewiesen, alle von Ihnen durchgeführten Änderungen zu ignorieren.

### 4.3 Die Prozedur `fuenf` editieren

Im nachfolgenden Beispiel können Sie sich im Umgang mit dem Editor üben, indem Sie die Prozedur `fuenf` aus Lektion 3 so ändern, daß sie um 36 Grad gedreht und zehnmal wiederholt wird. Ferner sollen Sie den Namen ändern, so daß der alte und der neue Name im Wortschatz von Dr. Logo vorhanden sind.

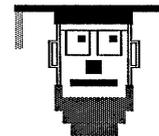
#### Übung 4-1: Ändern der Prozedur `fuenf`

Ihre Angaben

```
?ed "fuenf
```

Erläuterung

Als erstes verwenden Sie den Editor. Die Prozedur erscheint auf dem ungeteilten Bildschirm.



### Dr. Logo's Leitfaden

Wenn Dr. Logo wie folgt reagiert:

```
I don't know how to fuenf
```

haben Sie vergessen, " vor `fuenf` einzutippen. Versuchen Sie es noch einmal.

Wenn Dr. Logo wie folgt reagiert:

```
I don't know how to ed"fuenf
```

haben Sie vergessen, ein Leerzeichen zwischen `ed` und `"fuenf` einzutippen.

```
to fuenf
repeat 5 [fd 40 rt 72]
end
```

```
to fuenf1
repeat 5 [fd 40 rt 72]
end
```

```
to fuenf1
repeat 10 [rt 36 repeat 5 [fd
40 rt 72]]
end
```

?fuenf1



Der Cursor erscheint rechts von fuenf. Tippen Sie 1, damit der neue Name fuenf1 lautet. Bewegen Sie den Cursor eine Zeile nach unten an den Anfang der Zeile direkt unter das r in repeat.

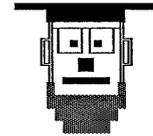
Geben Sie die nachfolgend gezeigte zusätzliche Anweisung ein. Der Editor fügt diese automatisch ein und verschiebt den Rest der Zeile nach rechts, um für die Einfügung Platz zu machen. Denken Sie daran, eine schließende eckige Klammer (] bzw. Ü) am Ende der Zeile anzufügen.

Überprüfen Sie die Prozedurdefinition noch einmal, um zu sehen, ob sie richtig ist. Wenn dies nicht der Fall sein sollte, machen Sie die nötigen Änderungen.

Wenn Sie zufrieden sind, drücken Sie Exit bzw. Alt- oder Control-C, damit die neue Definition angenommen und der Editor verlassen wird. Sie kehren zu Dr. Logo zurück.

Probieren Sie die Prozedur nun noch einmal aus.

Es funktioniert!



## Dr. Logo's Leitfaden

Wenn eine Prozedur nach erfolgter Änderung nicht arbeitet, bedeutet dies, daß immer noch irgend etwas nicht stimmt. Verwenden Sie den Editor noch einmal, und machen Sie solange Änderungen, bis die Prozedur läuft. Die Hilfe-Meldungen von Dr. Logo zeigen Ihnen, was geändert werden sollte. Wenn Sie mehr Erfahrung bei der Programmierung mit Dr. Logo haben, werden Sie mehr Zeit damit verbringen und mehr Freude daran haben, Prozeduren zu ändern.

Wenn Sie wollen, daß sich Dr. Logo an fuenf und fuenf1 noch erinnern kann, nachdem Sie Ihren Computer ausgeschaltet haben, lesen die nächste Lektion durch, in der Sie lernen werden, wie Sie Prozeduren sichern.

## 4.4 Zusammenfassung

Was Sie Neues gelernt haben:

Sie können die Definition von Prozeduren mit Hilfe des Texteditors ändern.

Neue Grundwörter:

Name	Eingabe
ed	Der Name der Prozedur, vor dem ein " steht.

Neue, von Ihnen definierte Prozeduren:

Name	Eingabe
fuenf1	keine

## 5 Prozeduren sichern

Dr. Logo erinnert sich an die von Ihnen definierten Wörter nur so lange, bis der Computer abgeschaltet wird. Wenn Sie Ihren Computer ausschalten und dann wieder einschalten, reagiert Dr. Logo nach Eintippen von `fuenf` wie folgt:

```
I don't know how to fuenf
```

Selbstverständlich gibt es aber eine Möglichkeit, Ihre neuen Prozeduren auf einer Diskette zu sichern, so daß Sie sie auch später verwenden können. In dieser Lektion wird dieser Vorgang erklärt.

### Lernziele

1. Sie sollen lernen, wie man Informationen in Dateien speichert.
2. Sie sollen den Inhalt Ihres Arbeitsspeichers überprüfen.
3. Sie sollen lernen, wie man etwas löscht.

### 5.1 Wie Daten gespeichert werden

Alle Prozeduren, die Sie seit dem letzten Einschalten des Computers geschrieben haben, bilden den Arbeitsspeicher. Dr. Logo vergißt alle diese Prozeduren, wenn Sie den Computer ausschalten, sofern Sie nicht den Inhalt des Arbeitsspeichers auf einer Diskette gesichert haben. Wenn Sie Ihren Arbeitsspeicher auf einer Diskette sichern, nennt man dies *Datei*.

Dr. Logo kennzeichnet jede Datei mit einem von Ihnen ausgewählten Namen, sofern dieser aus acht oder weniger Zeichen besteht. Nachfolgend einige Beispiele für Dateinamen, die Sie später vielleicht einmal erstellen wollen:

```
zeichng  
wrtspiel  
telbuch
```

Es gibt zwei Grundwörter, die Sie für die Sicherung Ihres Arbeitsspeichers als Datei verwenden und später einsetzen können:

**save** mit der Eingabe eines Dateinamens. Dadurch wird Dr. Logo angewiesen, den gesamten Arbeitsspeicher in einer Datei mit diesem Namen zu sichern.

**load** mit ebenfalls einer Eingabe, einem Dateinamen. Dies ist ein Hinweis für Dr. Logo, diese Datei zu lesen und sie in den Arbeitsspeicher Ihres Rechners zu laden.

Wenn Sie das vorliegende Handbuch durcharbeiten und dabei den Arbeitsspeicher sichern wollen, bevor Sie den Computer ausschalten, müssen Sie folgende Schritte ausführen:

1. Nehmen Sie eine saubere leere Diskette, auf der Sie den Dr.-Logo-Arbeitsspeicher speichern wollen. Wenn die neue Diskette bereits initialisiert ist, machen Sie bei Schritt 2 weiter. Wenn die Diskette noch nicht initialisiert ist, müssen Sie sie formatieren, damit Sie sie lesen und Daten darauf schreiben können.

Tippen Sie `bye` nach dem `?`-Prompt ein.

Verwenden Sie das Formatierprogramm `DISCKIT3 (CPC)` bzw. `DISCKIT (Joyce)`, um Ihre Diskette damit zu formatieren.

Laden Sie Dr. Logo erneut.

2. Wenn Sie eine separate Daten-Diskette benutzen wollen, nehmen Sie die Dr.-Logo-Systemdiskette aus dem Laufwerk heraus - wenn nicht, überspringen Sie Punkt 3.
3. Legen Sie die Daten-Diskette in das Laufwerk ein.
4. Sobald das `?`-Prompt erscheint, tippen Sie folgendes ein:

```
save "test
↑
Leerzeichen
```

Tippen Sie die Anführungszeichen *vor* den Dateinamen, jedoch *nicht* danach. Der Computer klickt und surrt. Das `?`-Prompt erscheint erneut, sobald die Daten auf einer Datei mit Namen `test` gesichert sind. Nehmen Sie die Diskette so lange nicht aus dem Laufwerk, bis das `?` erneut erscheint.

Jedesmal nach dem erneuten Einschalten des Computers nehmen Sie, wenn Sie eine separate Daten-Diskette angelegt haben, die Dr.-Logo-Systemdiskette aus dem Laufwerk, legen diese Diskette mit der Datei ein und tippen nach dem `?`-Prompt folgendes ein:

```
load "test
```

Dadurch wird Dr. Logo angewiesen, den Arbeitsspeicher mit allen in der Datei enthaltenen Daten zu laden. Die Namen aller Prozeduren in dieser Datei werden am Bildschirm angezeigt, sobald sie von der Diskette in den Arbeitsspeicher kopiert werden.

Nachfolgend eine Zusammenfassung der Regeln für Dateien:

- Sie sollten Dateien auf einer eigenen, initialisierten Diskette und nicht auf der Dr. Logo-Systemdiskette speichern.
- Sie legen Dateien an, indem Sie `save "Dateiname` eintippen. Dabei ist es egal, ob Sie den Dateinamen klein oder groß schreiben.
- Sie laden die Daten der Dateien in den Arbeitsspeicher durch Eintippen von `load "Dateiname`.
- Dateien bleiben so lange auf einer Diskette gespeichert, bis Sie diese löschen oder die Diskette beschädigt wird.
- Durch Eintippen von `save "Dateiname` wird jede Prozedur und der entsprechende Name im Arbeitsspeicher gesichert, einschließlich aller anderen Dateien, die Sie nach dem Einschalten Ihres Computers geladen haben.

**Anmerkung:** Dieser letzte Punkt ist ein bißchen verwirrend. Wenn Sie mehrere Prozeduren und Namen im Arbeitsspeicher haben, können Sie die Prozeduren nicht einzeln sichern. Wenn Sie `save` eintippen und danach einen Dateinamen, wird alles aus dem Arbeitsspeicher gespeichert.

- Wenn auf der Diskette bereits eine Datei mit dem angegebenen Namen besteht, zeigt `save` folgende Meldung an:

```
File Dateiname already exists
```

Wählen Sie einen anderen Namen für die neue Datei, löschen Sie die alte Datei mit Hilfe des Kommandos `erasefile` oder benennen Sie die alte Datei mit Hilfe des Kommandos `changeif` um, und sichern Sie dann die neue Datei. Weitere Einzelheiten über `erasefile` und `changeif` finden Sie im Dr.-Logo-Lexikon.

## 5.2 Der Arbeitsspeicher

Wenn Sie den Inhalt Ihres Arbeitsspeichers überprüfen wollen, können Sie die Grundwörter `pots` und `po` verwenden.

Durch Eintippen von `pots` werden die Namen aller Prozeduren im Arbeitsspeicher angezeigt. Sie können diese Kommandos verwenden, um Ihre Arbeit zu überprüfen oder um Ihrem Gedächtnis bei der Suche nach einem bestimmten Prozedurnamen auf die Sprünge zu helfen. Wenn Sie beispielsweise fünf geschrieben haben und Dr. Logo wie folgt reagiert hat:

```
I don't know how to funf
```

könnten Sie `pots` eintippen, um alle Prozedurnamen im Arbeitsspeicher zu überprüfen und den richtigen Namen, nämlich `fuenf`, finden.

```
?pots
to quadrat
to fuenf
to fuenf1
```

Das Eintippen von `po` und Prozedurname zeigt die Definition dieser Prozedur an. Sie können dieses Kommando dazu verwenden, eine Prozedurdefinition zu überprüfen, ohne Änderungen daran auszuführen.

```
?po "fuenf1
to fuenf1
repeat 10 [rt 36 fuenf]
end
?
```

## 5.3 Wie Informationen gelöscht werden

Dr. Logo löscht die Prozeduren in Ihrem Arbeitsspeicher, wenn Sie die Grundwörter `er` oder `erall` eintippen

`er` Mit `er` und danach einem Leerzeichen und einem Prozedurnamen, vor dem ein Anführungszeichen steht, wird die entsprechende Prozedurdefinition aus dem Arbeitsspeicher gelöscht. Verwenden Sie `er` zum Löschen einer Prozedur, wenn Sie diese nicht mehr benötigen. Beispiel:

```
?er "fuenf
```

`erall` Mit `erall`, sofern dies allein eintippt wird, werden alle Namen und Prozeduren aus dem Arbeitsspeicher gelöscht. Verwenden Sie dieses Grundwort, um den Arbeitsspeicher zu löschen, wenn keine der dort vorhandenen Prozeduren gespeichert werden soll.

## 5.4 Zusammenfassung

Was Sie Neues gelernt haben:

- o Dr. Logo *vergißt* neue von Ihnen geschriebene Prozeduren, wenn Sie den Computer ausschalten, es sei denn, Sie haben diese Prozeduren auf einer Disketten-datei gesichert.
- o Sie können den Inhalt Ihres Arbeitsspeichers auf einer initialisierten Diskette speichern und später wieder laden.

Neue Grundwörter:

Name	Eingabe
<code>save</code>	1 Dateiname mit 8 oder weniger Zeichen (" davor eintippen!)
<code>load</code>	1 Dateiname mit acht oder weniger Zeichen (" davor!)
<code>pots</code>	keine
<code>po</code>	1 Prozedurname (" davor eintippen!)
<code>er</code>	1 Prozedurname (" davor eintippen!)
<code>erall</code>	keine

## 6 Prozeduren mit Eingaben

Mit Dr. Logo können Sie Ihre Prozeduren flexibel gestalten, indem Sie sie so schreiben, daß sie Eingaben akzeptieren, wie dies auch bei bestimmten Grundwörtern der Fall ist. In dieser Lektion sollen Sie lernen, wie man Prozeduren mit Eingaben schreibt.

### Lernziele

1. Sie sollen einige Regeln über Eingaben beherrschen.
2. Sie sollen Prozeduren schreiben, mit denen Sie verschieden große Zeichnungen erstellen können.

### 6.1 Wie man Eingaben verwendet

Grundwörter, für die keine Eingaben notwendig sind (wie z.B. `cs` oder `ht`), führen immer genau die gleiche Operation durch, während Grundwörter, die eine Eingabe benötigen, verschiedene, dem von Ihnen angegebenen Wert entsprechende Dinge tun. Beispielsweise führt `fd 50` zu einem anderen Ergebnis als `fd 90`, und `pr [HALLO]` führt zu einem anderen Ergebnis als `pr [Auf Wiedersehen]`.

Bisher haben die von Ihnen definierten Prozeduren keine Eingabe verwendet. Wenn Sie den Prozedurnamen `quadrat` eingeben, wird immer ein Quadrat in der Größe gezeichnet, die Sie in der Prozedurdefinition angegeben haben; wenn Sie den Prozedurnamen `fuenf` eingeben, wird ein Fünfeck in genau der Größe gezeichnet, wie in der Prozedurdefinition angegeben.

Sie können variable Prozeduren schreiben, indem Sie einen *Platzhalter* für die Eingabe nach dem Prozedurnamen einfügen und einen weiteren Platzhalter in der Prozedurdefinition an die Stelle setzen, an der die Eingabe erfolgen wird. Wenn Sie dann die Prozedur ausführen, geben Sie eine Eingabe nach dem Prozedurnamen an der Stelle an, an der der Platzhalter stand.

Sie setzen einen Platzhalter dadurch, daß Sie einen Doppelpunkt (`:`) eintippen und danach ein Wort, das die Eingabe rechts davon beschreibt. Machen Sie kein Leerzeichen zwischen dem Doppelpunkt und dem Wort.

Versuchen Sie einmal folgendes:

Leerzeichen



```
?to treppe :stufe
>repeat :stufe [fd 5 rt 90 fd 5 lt 90]
>end
treppe defined
?
```

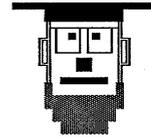
Wenn Sie jetzt `treppe` ausführen, müssen Sie eine Ziffer danach eintippen, die angibt, wie oft das Kommando zu wiederholen ist! (Sie müssen den Doppelpunkt nicht eintippen, wenn Sie die Prozedur ausführen.)



```
?treppe 2
?
```



```
?cs
?treppe 9
?
```



## Dr. Logo's Leitfaden

Wenn folgende Meldung angezeigt wird:

```
Not enough inputs to treppe in treppe: treppe
```

haben Sie nach dem Kommando `treppe` keine Eingabe eingetippt. Denken Sie daran, daß Prozeduren, die Platzhalter enthalten, zur Ausführung Eingaben benötigen.

## 6.2 Prozeduren für Zeichnungen verschiedener Größe

Das folgende Beispiel zeigt, wie Sie eine Prozedur ähnlich der Prozedur `quadrat` schreiben, die als Eingabe die Angabe der Seitenlänge benötigt.

### Übung 6-1: Prozeduren mit Eingaben schreiben

Ihre Angaben	Erläuterung
<code>?to kästchen :seite</code>	Die Prozedur <code>kästchen</code> benötigt eine Eingabe: die Seitenlänge
<code>&gt;repeat 4 [fd :seite rt 90]</code>	Der Igel rückt um so viele Einheiten vorwärts, wie Sie nach dem Prozedurnamen eintippen.
<code>&gt;end</code> <code>kästchen defined</code>	Probieren Sie jetzt die Prozedur aus:
<code>?kästchen</code> <code>Not enough inputs to kästchen!</code> <code>in kästchen: kästchen</code>	Die Prozedur <code>kästchen</code> läuft nur, wenn Sie die Seitenlänge nach dem Prozedurnamen eingeben!

**?kästchen 30**

Der Igel zeichnet ein Quadrat mit einer Seitenlänge von 30 Einheiten.

**?kästchen 40**

Der Igel zeichnet ein Quadrat mit einer Seitenlänge von 40 Einheiten.



Sie haben jetzt `kästchen` definiert; die Prozedur wird in den Wortschatz von Dr. Logo übernommen, und Sie können diese in anderen Prozedurdefinitionen verwenden.

Im folgenden Beispiel definieren Sie andere Prozeduren, die `kästchen` und die Eingabe für `kästchen` als Teil ihrer Definition verwenden.

### Übung 6-2: Weitere Prozeduren schreiben, die Zeichnungen verschiedener Größe erstellen

Ihre Angaben	Erläuterung
<code>?to spinnetz :seite</code>	<code>spinnetz</code> benötigt eine Eingabe: die Seitenlänge
<code>&gt;repeat 36 [kästchen :seite r! t 10] &gt;end spinnetz defined</code>	Der Igel verwendet die Eingabe, um ein Quadrat zu zeichnen (die Prozedur <code>kästchen</code> aus Übung 6-1 wird dafür verwendet), dreht sich dann 10 Einheiten nach rechts und wiederholt diese Liste der Anweisungen 36 mal.
<code>?spinnetz 40</code>	Probieren Sie <code>spinnetz</code> jetzt aus.  Der Igel führt die Prozedur <code>spinnetz</code> aus, wobei die Seiten des Quadrats jeweils 40 Einheiten lang sind.



Wiederholen Sie die Ausführung von `spinnetz`, ohne den Bildschirm zu löschen, so daß die zweite Zeichnung über der ersten liegt.

**?spinnetz 80**

Der Igel zeichnet das gleiche Muster, allerdings zweimal größer!

**?spinnetz**

Not enough inputs to spinnetz!  
in spinnetz: spinnetz

Dr. Logo weiß nicht, wie groß die Kästchen sein sollen, da die Eingabe fehlt.

Sie können auch eine Prozedur schreiben, die drei Eingaben benötigt, so daß Dr. Logo angewiesen wird, drei derartige Zeichnungen in verschiedener Größe zu entwerfen. Nennen Sie diese Prozedur `facette`, da das Resultat genauso aussieht.

```
?to facette :X :Y :Z
>spinnetz :X
>spinnetz :Y
>spinnetz :Z
>end
facette defined
```

`facette` benötigt drei Eingaben (Sie können beliebige Platzhalter verwenden; X, Y und Z sind leicht einzutippen). Jede Eingabe bestimmt die Länge der Seite der Kästchen für eine vollständige Ausführung von `spinnetz`.

Führen Sie die Prozedur jetzt aus.

**?facette 30 40 70**

Der Igel zeichnet ein `spinnetz` mit Kästchen von 30 Einheiten Länge, eines mit Kästchen von 40 Einheiten Länge und eines mit Kästchen von 70 Einheiten Länge.



## 6.3 Zusammenfassung

Was Sie Neues gelernt haben:

- o Eine Prozedur schreiben, die verschiedene Eingaben akzeptiert (eine variable Prozedur), wobei ein Platzhalter bzw. die Eingabe nach dem Prozedurnamen und in der Definition an der vorgesehenen Stelle eingefügt wird.

- o Einen Platzhalter einfügen, indem ein Doppelpunkt und unmittelbar danach ein beschreibendes Wort für die Eingabeart eingetippt wird.

Neue, von Ihnen definierte Prozeduren:

Name	Anzahl der Eingaben
terppe	1 Zahl
kästchen	1 Zahl
spinnetz	1 Zahl
facette	3 Zahlen

## 7 Programme erstellen

Eine Grundidee für das Programmieren mit Dr. Logo besteht darin, Prozeduren zu schreiben und diese als Bausteine für andere Prozeduren zu verwenden. Die Prozedur *spinnetz*, die Sie in der vorangegangenen Lektion geschrieben haben, wurde beispielsweise ein Baustein für die Prozedur *facette*. In der folgenden Lektion sollen Sie lernen, Prozeduren zu schreiben, die das Ergebnis anderer Prozeduren verwenden.

### Lernziele

1. Sie sollen lernen, was Ausgabe bedeutet.
2. Sie sollen lernen, eine Prozedur zu schreiben, die andere Prozeduren verwendet.

### 7.1 Was sind Ausgaben

Einige Grundwörter erzielen eine Wirkung, wie z.B. *fd*, *ht* oder *rt*, während andere Grundwörter zu einem Ergebnis führen, wie z.B. *+* und *\**. Wenn Prozeduren bei Dr. Logo Ergebnisse erstellen, müssen Sie Dr. Logo mitteilen, was damit geschehen soll. Wenn Sie sich jede Prozedur als eine Person an einem Montageband vorstellen, wird Ihnen klar, daß jede Prozedur ein eigenes Ergebnis erstellt, bis sie angewiesen wird, das Ergebnis an die nächste Prozedur weiterzugeben bzw. *auszugeben*.

Um Dr. Logo anzuweisen, ein Ergebnis auszugeben, müssen Sie das Grundwort *op* verwenden.

*op* (output) weist Dr. Logo an, das Ergebnis einer Prozedur auszugeben, so daß eine andere Prozedur es verwenden kann. Die Prozedur hört an diesem Punkt auf; sobald ein Ergebnis ausgegeben ist, ist die Prozedur beendet.

Sehen Sie sich die folgende Definition für die Prozedur *add* an:

```
?to add :n
>:n + 1
>end
add defined
?add 2
I don't know what to do with 2 in add: :n + 1
```

Dr. Logo wußte nicht, was er mit 2 machen soll - ob es gedruckt werden soll, ob der Igel zwei Schritte bewegt werden soll oder was auch immer.

Tippen Sie jetzt ed ein, und drücken Sie auf die Return-Taste, um die Prozedur add zu ändern. Fügen Sie das Wort op vor :n + 1 ein.

```
to add :n
op :n + 1
end
add defined
?add 2
3
```

In der ersten Prozedur wußte Dr. Logo nicht, was mit der Eingabe geschehen sollte. In der zweiten Prozedur wird Dr. Logo durch das Grundwort op angewiesen, das Ergebnis am Bildschirm anzuzeigen. Wenn eine Prozedur eine Ausgabe erstellt, können Sie diese Ausgabe als Eingabe für andere Prozeduren verwenden!

Im folgenden Beispiel wird gezeigt, wie Sie eine Prozedur schreiben, die die Ausgabe einer anderen Prozedur verwendet.

#### Übung 7-1: Die Zeit zwischen zwei Jahreszahlen errechnen

Ihre Angaben	Erläuterung
<pre>?to zeitraum :damals :heute &gt;op :damals - :heute &gt;end zeitraum defined ?</pre>	Zunächst können Sie eine Prozedur schreiben, die die Anzahl der Jahre zwischen zwei vorgegebenen Jahreszahlen errechnet.
<pre>?to tage :damals :heute &gt;op 365 * zeitraum :damals ! :heute &gt;end tage defined ?</pre>	Jetzt können Sie das Grundwort zeitraum als Baustein einer anderen Prozedur verwenden, welche die Anzahl der Tage zwischen zwei vorgegebenen Jahreszahlen errechnet. Dieses Programm nimmt an, daß Sie jeweils vom gleichen Tag im entsprechenden Jahr ausgehen.
<pre>?tage 2000 1983 6205</pre>	Wieviel Tage liegen zwischen 1983 und dem Jahr 2000?

## 7.2 Eine Prozedur unter Verwendung anderer Prozeduren schreiben

Der Reiz beim Programmieren mit Dr. Logo besteht darin, daß Sie - sobald Sie eine Prozedur definiert haben - Zeit sparen, indem Sie diese Prozedur als Baustein für an-

dere Prozeduren verwenden können. Im folgenden Beispiel wird gezeigt, wie zwei kleine Prozeduren für die Erstellung einer größeren Prozedur benutzt werden.

#### Übung 7-2: Eine Prozedur unter Verwendung anderer Prozeduren schreiben

Ihre Angaben	Erläuterung
<pre>?to rahmen &gt;repeat 4 [fd 30 rt 90] &gt;end rahmen defined  ?rahmen</pre>	Um ein Haus zu zeichnen, benötigen Sie zwei Elemente: einen Rahmen und ein Dach.



```
?to dach
>rt 30 fd 30
>rt 120 fd 30
>end
dach defined
?cs
?dach
```



```
?to haus
>rahmen
>pu fd 30 pd
>dach
>end
haus defined
```

Das Dach sieht wie der obere Teil eines Dreiecks aus.

Schreiben Sie mit Hilfe dieser beiden Prozeduren nun ein Programm für ein Haus. Sie müssen den Igel in die richtige Position bringen, damit Sie nach der Zeichnung des Rahmens mit der Zeichnung des Daches beginnen können.

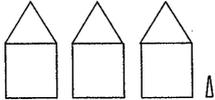
?cs  
?haus



```
?to reihe
>repeat 3
> [pd haus pu
> rt 30 fd 30
> lt 90 fd 20 lt 90]
>end
reihe defined
```

Jetzt können Sie die Prozedur `haus` in einem Programm verwenden, das mehrere Häuser zeichnet. Sie müssen die Position des Igels nach Zeichnung jedes Hauses so setzen, daß sie zum Zeichnen eines weiteren Hauses den richtigen Ausgangspunkt hat.

?reihe



Sie können eine lange Zeile in Ihrer Prozedur in mehrere Zeilen aufsplitten und diese eingerückt schreiben. Dr. Logo liest die mittleren drei Zeilen der Prozedur `reihe` wie eine Zeile.

### 7.3. Zusammenfassung

Was Sie Neues gelernt haben:

- o In den Prozeduren müssen Sie Dr. Logo anweisen, was mit dem Ergebnis einer Operation geschehen soll. Sie können dies durch Verwendung des Grundwortes `op tun`.
- o Sie können von Ihnen definierte Prozeduren als Bausteine für größere Prozeduren verwenden.

Neue Grundwörter:

Name	Anzahl der Eingaben
<code>op</code>	1

Neue, von Ihnen definierte Prozeduren:

Name	Anzahl der Eingaben
<code>add</code>	1 Zahl
<code>zeitraum</code>	2 Zahlen
<code>tage</code>	2 Zahlen
<code>rahmen</code>	keine
<code>dach</code>	keine
<code>haus</code>	keine
<code>reihe</code>	keine

## 8 Prozeduren, die sich selbst aufrufen

Manchmal möchten Sie, daß sich eine Prozedur für eine Reihe von Eingaben selbst wiederholt, so daß Sie die Prozedur nicht mehrmals ausführen müssen. Sie können Dr. Logo anweisen, dies unter Verwendung des Prozedurnamens in der eigenen Definition zu tun. In der folgenden Lektion sollen Sie lernen, wie Sie Prozeduren schreiben, die sich selbst immer wieder aufrufen.

### Lernziele

1. Sie sollen eine Prozedur schreiben, die sich selbst in ihrer Definition aufruft.
2. Sie sollen lernen, wie Prozeduren beendet werden.
3. Sie sollen eine Prozedur schreiben, die aus einer Liste von Zahlen ein Balkendiagramm zeichnet.

### 8.1 Eine Prozedur schreiben, die sich selbst aufruft

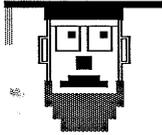
*Ein Kreis ist kreisförmig.* Das wäre keine gute Definition für einen Kreis in einem Lexikon, da die Definition Sie in einer unendlichen Schleife immer wieder zurück auf das zu definierende Wort führt.

Bei Dr. Logo ist es nicht nur grammatikalisch richtig, einen Prozedurnamen in dessen Definition zu verwenden, sondern es ist dazu noch eine sehr gute Technik, um eine Prozedur am Laufen zu halten.

Nachfolgend eine Möglichkeit, eine Prozedur für ein Quadrat zu schreiben, die sich selbst immer wieder verwendet. (Drücken Sie Control-G bzw. Stop, um die Prozedur zu beenden.)

```
?to quadrat :länge
>fd :länge rt 90
>quadrat :länge
>end
quadrat definiert
?
```

Jedesmal, wenn die Prozedur ausgeführt wird, springt Dr. Logo an den Anfang zurück und beginnt, die Definition erneut auszuführen!



## Dr. Logo's Leitfaden

Wenn Sie eine Meldung ähnlich der folgenden sehen:

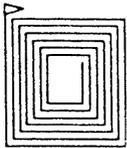
Out of space

tippen Sie das Grundwort `recycle` ein und warten, bis das `?`-Prompt erneut erscheint. Dadurch wird Dr. Logo angewiesen, alle Prozeduren im Arbeitsspeicher neu zu organisieren, damit mehr Platz entsteht.

Der Vorteil beim Schreiben von Prozeduren, die sich selbst aufrufen, besteht darin, daß Sie diese editieren können, so daß die Eingabe jedesmal anders ist, wenn Dr. Logo einen neuen Zyklus beginnt.

Sehen Sie sich zum Beispiel die nachfolgende Prozedur an, und versuchen Sie, sich die zu zeichnende Figur vorzustellen, bevor Sie die Prozedur ausführen. (Denken Sie daran, daß Sie mit Control-G abbrechen können.)

```
?to spirale :länge
>fd :länge rt 90
>spirale :länge + 3
>end
spirale definiert
?
```



```
?spirale 40
```

```
Stopped! in spirale: spirale
```

```
?
```

## 8.2 Wie Prozeduren beendet werden

Es gibt eine Möglichkeit, ein *Stopp-Kommando* in die Prozeduren einzubauen, so daß Sie Control-G nicht drücken müssen. Sie können eine Zeile in der Prozedur schreiben, die Dr. Logo anweist, einen bestimmten Sachverhalt zu untersuchen (beispielsweise die Länge einer Seite) und eine bestimmte Aktion auszuführen, wenn dieser Sachverhalt zutrifft und eine andere, wenn er nicht zutrifft.

Sie können diese Urteilsfähigkeit von Dr. Logo mit den Grundwörtern `if` und `stop` verwenden und dadurch die Beendigung einer Prozedur festlegen.

`if` benötigt zwei Eingaben: eine Wahr- oder Falsch-Anweisung und eine Anweisung in eckigen Klammern. Dr. Logo befolgt die Anweisung dann, wenn sie wahr ist (TRUE).



```
?if 15 = 15 [fd 10]
?if 15 < 10 [fd 10]
?if 15 < 10 [fd 10] bk 50
?
```

In der letzten Zeile bewegte Dr. Logo den Igel keine 10 Schritte mehr vorwärts, da die Anweisung nicht *wahr* war. Statt dessen ging Dr. Logo weiter zur nächsten Anweisung und bewegte den Igel 50 Schritte zurück.

`stop` weist Dr. Logo an, die Prozedur zu beenden.

Im folgenden Beispiel wird gezeigt, wie `if` und `stop` zusammen verwendet werden, um die Prozedur `vielspi` zu beenden, sobald die Seiten eine bestimmte Länge erreicht haben.

## Übung 8-1: Die Prozedur vielspi

Ihre Angaben

`?to vielspi :länge :winkel``>fd :länge``>rt :winkel``>if :länge > 90 [stop]``>vielspi :länge + 3 :winkel``>end  
vielspi definiert``?vielspi 1 120``?cs  
?vielspi 5 45`

Erläuterung

vielspi benötigt zwei Eingaben: nämlich :länge und :winkel

Der Igel geht vorwärts :länge.

Der Igel dreht sich nach rechts :winkel.

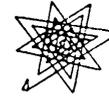
Wenn die Größe der Seite 90 Einheiten überschreitet, wird die Prozedur beendet.

Der Igel sucht erneut die Definition von vielspi und erhöht die Eingabe :länge um 3.

Testen Sie jetzt Ihre Prozedur unter Verwendung verschiedener Werte als Eingaben. Denken Sie daran, daß zwei Platzhalter in der Prozedur-Definition enthalten sind und Sie daher unbedingt zwei Eingaben eintippen müssen, damit die Prozedur ausgeführt wird.

Der Igel dreht sich jedesmal um 120 Grad, sobald er eine Seite gezeichnet hat.

Der Igel dreht sich jedesmal um 45 Grad, wenn er eine neue Seite gezeichnet hat.

`?cs``?vielspi 1 135`

Versuchen Sie, vielspi unter Verwendung eigener Eingaben auszuführen.

### 8.3 Ein Balkendiagramm zeichnen

Es gibt viele wirklichkeitsnahe Situationen, in denen Sie eine Prozedur schreiben wollen, die sich selbst aufruft. Zum Beispiel: Überlegen Sie, wie Sie ein Programm schreiben würden, das ein Balkendiagramm durch Verwendung einer Liste aus Zahlen so aufzeichnet, daß Sie die verschiedenen Werte vergleichen könnten. Zunächst müßten Sie die erste Ziffer grafisch darstellen, dann die nächste usw., bis die Liste abgearbeitet wäre.

Es gibt ein paar ganz spezielle Grundwörter, die Sie für das Arbeiten mit Listen verwenden können:

**first** benötigt eine Eingabe, nämlich eine Liste. Dadurch wird Dr. Logo angewiesen, das erste Element in der Liste auszudrucken.

**last** benötigt auch eine Liste als Eingabe. Dadurch wird Dr. Logo angewiesen, das letzte Element in der Liste auszudrucken.

`?first [12 3 6]``12``?last [12 3 6]``6`

**bf** (but first) benötigt eine Liste als Eingabe. Dadurch wird Dr. Logo angewiesen, die Liste ohne das erste Element auszudrucken.

**bl** (but last) benötigt eine Liste als Eingabe. Dadurch wird Dr. Logo angewiesen, die Liste ohne das letzte Element auszugeben.

`?bf [12 3 6]``[3 6]``?bl [12 3 6]``[12 3]`

Sie können diese Grundwörter auch zusammen verwenden, um eine beliebige Kombination von Elementen aus einer Liste herauszuziehen.

```
?first bf [1 2 3 4 5]
```

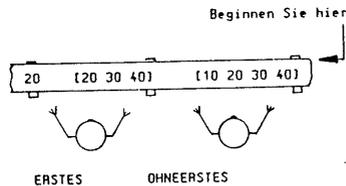
```
2
```

```
?last bl [1 2 3 4 5]
```

```
4
```

Im ersten Beispiel nahm Dr. Logo `bf` der Liste, d.h. `[2 3 4 5]`, und gab dann das erste Element dieser Liste aus.

Im zweiten Beispiel nahm Dr. Logo `bl` der Liste, d. h. `[1 2 3 4]`, und gab dann das letzte Element dieser Liste aus.



Sie können sich jedes Grundwort in einer Zeile als einen von Dr. Logos Helfern an einem Montageband vorstellen. Das Grundwort, das am nächsten zur Eingabe steht, führt seine Aufgabe aus, gibt dann das Ergebnis an das nächste Grundwort weiter, damit dieses seine Aktion ausführen kann usw., das ganze Band entlang, bis das fertige Produkt ausgeliefert wird. Der Schlüssel des Ganzen besteht darin, daß Sie von rechts nach links lesen müssen.

Verwenden Sie jetzt diese Grundwörter für Listen, und schreiben Sie eine Prozedur, die eine Liste mit Zahlen grafisch darstellt.

### Übung 8-2 - Teil 1: Eine Prozedur zum Zeichnen von Balken schreiben

Ihre Angaben

```
?to balken :zahl
```

```
>fd :zahl
```

```
>bk :zahl
```

```
>rt 90 fd 10 lt 90
```

```
>end
```

```
balken definiert
```

Erläuterung

Schreiben Sie zunächst eine einfache Prozedur, mit der Balken gezeichnet werden können. Die Prozedur `balken` benötigt eine Eingabe: die Zahl, die grafisch dargestellt werden soll.

Der Igel wird vorwärts und rückwärts um die in der Eingabe gemachte Anzahl von Einheiten bewegt.

Der Igel wird 10 Einheiten nach rechts bewegt, so daß er den nächsten Balken zeichnen kann.

?balken 30



?balken 40



Probieren Sie jetzt die Prozedur `balken` aus.

Um eine andere Prozedur zu definieren, die eine Liste mit Zahlen als Eingabe akzeptiert, benötigen Sie ein weiteres Grundwort, mit dem Dr. Logo prüft, ob die Liste leer ist.

`empty`

weist Dr. Logo an zu überprüfen, ob die danach stehende Eingabe leer ist. Es wird `TRUE` ausgedruckt, wenn sie leer ist und `FALSE`, wenn dies nicht der Fall ist.

```
?empty[ ]
```

```
TRUE
```

```
?empty[1 2 3]
```

```
FALSE
```

Definieren Sie jetzt eine Prozedur, die eine Liste aus Zahlen grafisch darstellt.

### Übung 8-2 - Teil 2: Eine Prozedur zum Zeichnen eines Balkendiagramms schreiben

Ihre Angaben

```
?to balkdia :liste
```

```
>if empty :liste [stop]
```

```
>balken first :liste
```

Erläuterung

Die Prozedur `balkdia` benötigt eine Eingabe: eine Liste aus Zahlen.

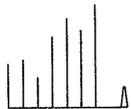
Wenn die Liste leer ist, beendet die Prozedur die Ausführung, und das Prompt erscheint erneut. Sonst geht Dr. Logo zur nächsten Anweisung über.

Dr. Logo nimmt das erste Element in der Liste und benutzt es als Eingabe für die Prozedur `balken`.

```
>balkdia bf :liste
```

```
>end
balkdia definiert
```

```
?cs
?balkdia [30 35 23 50 60 53
71]
```



## 8.4 Zusammenfassung

Was Sie Neues gelernt haben:

- o Sie können einen Prozedurnamen in die Definition dieser Prozedur aufnehmen.
- o Prozeduren, die sich selbst in ihren Definitionen enthalten, wiederholen sich endlos lange, bis Sie Control-G bzw. Stop drücken oder ein Stopp-Kommando in die Definition einfügen.

Neue Grundwörter:

Name	Anzahl der Eingaben
if	2 (eine Gleichung und eine Liste mit Anweisungen in eckigen Klammern)
stop	keine
recycle	keine
first	1 Liste
last	1 Liste
bf	1 Liste
bl	1 Liste
empty	1 Liste

Dr. Logo führt die Prozedur balkdia erneut aus und verwendet dafür bf in der Liste als Eingabe.

Testen Sie jetzt die Prozedur balkdia, indem Sie eine beliebige Liste mit Ziffern als Eingabe verwenden. (Dies könnten Lagerpreise, Anzahl von Sonnentagen zwischen Regentagen oder irgend etwas anderes sein.)

Neue, von Ihnen definierte Prozeduren:

Name	Anzahl der Eingaben
spirale	1 Zahl
vielspi	2 Zahlen
balken	1 Zahl
balkdia	1 Liste

## 9 Die Verwendung von Wörtern

Alle Wörter, die Sie bisher in den Prozeduren von Dr. Logo verwendet haben, waren Prozedurnamen. Es waren Anweisungen für Dr. Logo, eine Aktion, wie z.B. `fd`, `list`, `bf` oder `quadrat` durchzuführen. Manchmal möchten Sie, daß Dr. Logo ein Wort oder die Änderung eines Wortes einfach nur anzeigt, ohne die Definition des Wortes auszuführen. Beispiel: Sie wollen Dr. Logo mitteilen, den Satz `fd`, `MARSCH!` auszugeben, ohne den Igel vorwärts zu bewegen. In der folgenden Lektion sollen Sie lernen, wie Sie Wörter eintippen, die einfach nur Wörter sind und keine Prozedurnamen.

### Lernziele

1. Sie sollen Wörter eintippen, die *nur* Wörter und keine Prozedurnamen sind.
2. Sie sollen lernen, wie Definitionen durch Benennung Wörtern zugeordnet werden.
3. Sie sollen ein Telefonverzeichnis anlegen.
4. Sie sollen lernen, was Merkmallisten sind.
5. Sie sollen Ihr Telefonbuch unter Verwendung von Merkmallisten erweitern.

### 9.1 Wörter, die nur Wörter sind

Bei Dr. Logo gibt es zwei Wortarten: Prozedurnamen und Objekte.

**Prozedurnmn** Prozedurnamen sind Wörter, die Dr. Logo anweisen, eine Aktion durchzuführen. Dies können Grundwörter sein, wie z.B. `fd` und `list`, oder von Ihnen definierte Wörter, wie z.B. `quadrat` oder `balkdia`.

**Objekte** sind Wörter, von denen Dr. Logo nicht weiß, was damit anzufangen ist. Es sind einfach *nur* Wörter, und Sie verwenden sie als Eingabe für die Prozeduren. Bisher haben Sie nur Listen und Ziffern als Objekte verwendet.

Es gibt einen bestimmten Weg, Wörter, die Objekte sind, so einzutippen, daß Dr. Logo nicht versucht, sie als Prozeduren zu verwenden. Tippen Sie einfach Anführungszeichen vor das Wort, wie Sie dies nach den Grundwörtern `ed`, `save` und `er` getan haben. Nehmen wir an, Sie wollen, daß Dr. Logo das Wort `rt` ausdrückt, ohne daß der Igel gedreht wird.



Im ersten Beispiel listet Dr. Logo nur die ersten beiden Elemente der Liste auf, da list ohne Klammern nur zwei Eingaben akzeptiert.

## 9.2 Dinge benennen

Manchmal ist es nützlich, mit Dr. Logo ein Wort oder eine Ziffer, an die Sie sich erinnern *wollen*, an ein Wort oder eine Ziffer anzuhängen, an die Sie sich erinnern *werden*. Zum Beispiel: Sie können die Namen von Leuten, die Sie oft anrufen, mit deren Telefonnummern in Verbindung bringen. Wenn Sie später den Namen eintippen, zeigt Dr. Logo die dazu gehörige Telefonnummer an. Sie können auch eine Liste benennen, so daß Dr. Logo die ganze Liste anzeigt, wenn Sie deren Namen eingeben. Dadurch müssen Sie sich nur an einen Namen erinnern, anstatt sich eine ganze Liste mit Objekten zu merken.

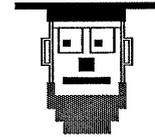
Es werden zwei Grundwörter für die Benennung verwendet:

**make** benötigt zwei Eingaben: die erste ist der neue Name, den Sie gewählt haben, die zweite ist dessen Definition. Die Definition wird mit dem Namen verknüpft.

**thing** (oder das Symbol :) benötigt eine Eingabe, nämlich einen Namen, den Sie bei Verwendung von make zugeordnet haben. Wenn Sie den Doppelpunkt : verwenden, dürfen Sie kein Leerzeichen zwischen dem Doppelpunkt und dem Namen einfügen. Dieses Grundwort weist Dr. Logo an, den dem Namen zugeordneten Wert anzuzeigen.

```
?make "haus "grün
?thing "haus
grün
?:haus
grün
?:grün
grün has no value
?make "grün "geht
?:grün
geht
```

So wie :seite ein Platzhalter für eine Ziffer in der Prozedur kästchen (Lektion 6) ist, so ist :grün ein Platzhalter für das Wort geht.



## Dr. Logo's Leitfaden

Setzen Sie das Symbol " (Anführungszeichen) nicht vor ein Wort, wenn Sie das Symbol : (Doppelpunkt) verwenden.

Nachfolgend verschiedene Möglichkeiten, wie man sich die Zeile

make "haus "grün

verdeutlichen kann.

- o haus ist der Name von grün
- o grün ist der Wert von haus
- o grün ist der Wert, der haus zugeordnet ist.

Das folgende Beispiel zeigt, wie Objekte benannt werden.

### Übung 9-2: Objekte benennen

Ihre Angaben

Erläuterung

```
?make "Susanne "Vorsitzende
?make "Peter "Schriftführer
```

Eine Möglichkeit, diese Zeilen laut zu lesen, ist: "Susanne ist Vorsitzende". "Peter ist Schriftführer".

```
?thing "Susanne
Vorsitzende
```

Wie heißt der Susanne zugeordnete Wert?

```
?:Peter
Schriftführer
```

Welcher Wert ist Peter zugeordnet?

```
?make "kennwort [Fühle die!
Kraft]
?:kennwort
[Fühle die Kraft]
```

Wie heißt der Wert, der kennwort zugeordnet ist?

?{list :Susanne :Peter) Die Susanne und Peter zugeordneten  
[Vorsitzende Schriftführer] Werte auflisten.

?make "mädchen [Gaby Sandra C! Das Benennen von Listen bietet eine ein-  
laudia] fache Möglichkeit, Verzeichnisse anzulegen.

?thing "mädchen  
[Gaby Sandra Claudia]

?make "geschenk [Tante.Lilo O! Sie können Punkte verwenden, um zwei  
nkel.Werner Leora.Hilb] zusammengehörende Wörter zu verknüpf-  
?:geschenk fen und Leerzeichen, um zwei verschiedene  
[Tante.Lilo Onkel.Werner Leor! Wörter zu trennen.  
a.Hilb]

### 9.3 Ein Telefonverzeichnis anlegen

Mit Hilfe der Benennung können Sie jetzt ein praktisch nutzbares Telefonverzeichnis anlegen!

#### Übung 9-3: Ein Telefonverzeichnis anlegen

Ihre Angaben	Erläuterung
?make "Monika 844582 ?:Monika 844582	Tippen Sie eine für Sie gültige Telefon- nummer ein. (Die nachfolgenden Angaben sind nur Beispiele.)
?make "Kurt 576761 ?make "Tante.Irene 611203 ?make "Städt.Klinik 597-1	Lesen Sie die Nummer unter Verwendung des Symbols : (Doppelpunkt).
	Geben Sie jetzt ein paar andere ein.

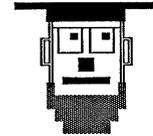
Um alles zur Anzeige zu bringen, was Sie benannt haben, können Sie das Grundwort pons verwenden.

pons steht für *Namen anzeigen* (print out names). Dadurch werden die von Ihnen benannten Werte in Satzform ausgegeben. Zum Beispiel:

```
"Peter ist ext39"
  ↑      ↑
  Name   Wert
```

Sie können dieses Grundwort dazu benutzen, die von Ihnen benannten Werte zu überprüfen. Dadurch können Sie Zeit sparen, wenn Sie beispielsweise einen Namen einer ganzen Liste zugeordnet haben, wie zum Beispiel geschenk.

```
?pons
haus is grün
grün is geht
kennwort is [Fühle die Kraft]
Monika is 844582
Kurt is 576761
Tante.Irene is 611203
geschenk is [Tante.Lilo Onkel.Werner Leora.Hilb]
?
```



### Dr. Logo's Leitfaden

Wenn Sie keinen der von Ihnen zugeordneten Namen sehen, haben Sie entweder Ihren Arbeitsspeicher nicht gesichert, d.h. Sie haben save "übung nicht eingetippt, bevor Sie den Computer ausgeschaltet haben, oder Sie haben ihn nach dem Wiedereinschalten des Computers nicht geladen. Wenn Sie vergessen haben, ihn zu laden, tippen Sie load "übung jetzt ein, und tippen Sie dann pons. Wenn Sie vergessen haben, Ihren Arbeitsspeicher zu sichern, müssen Sie alles neu eintippen, was Sie gesichert haben wollen. Vergewissern Sie sich, daß Sie dieses Mal save "übung eingetippt haben!

### 9.4 Merkmalliste

Es ist oft notwendig, mehrere Werte anstelle von nur einem Wert einem Objekt zuzuordnen. Beispiel: Sie wollen die Telefonnummern, Adressen und Geburtstage von Freunden oder die Position der Mannschaften in der Bundesliga und die Torverhältnisse festhalten.

Bei Dr. Logo werden die Beschreibungen, die Sie für eine Person oder ein Objekt verwenden, Merkmale genannt. Zum Beispiel: Die Merkmale eines Buches können den Autor, die Art (Roman/Gedicht) und die Anzahl der Seiten sein. Die Werte dieser Merkmale für das Buch Huckleberry Finn von Mark Twain könnten dann *Mark Twain, Roman* und *215* sein. Ähnlich könnten die Merkmale einer Briefmarke in Ihrer Sammlung das Land, das Ausgabejahr, das Datum des Erwerbs und den Wert umfassen. Die entsprechenden Werte wären z.B. *USA, 1963, 12/82* und *1.50*. Eine Reihe von Merkmalen und Werten, die einem Objekt zugeordnet sind, werden Merkmalliste genannt.

### Merkmalliste für Bücher

Objekt	Merkmale		
	Autor	Art	Seiten
Huckleberry Finn	Twain	Roman	215
Robinson Crusoe	Defoe	Roman	310

Bei Dr. Logo ist jedem Wort eine Merkmalliste zugeordnet; diese ist bei Wörtern, die Sie nicht benannt haben, leer und kann viele Merkmale und zugehörige Werte für Wörter enthalten, die Sie benannt haben.

Sie können die Grundwörter `pprop`, `gprop` und `plist` für die Arbeit mit Merkmallisten verwenden.

- pprop** (put property) steht für das Zuordnen eines Merkmals und dessen Wert an ein Objekt. Es benötigt drei Eingaben: ein Objekt, einen Merkmal-Namen und einen Wert.
- gprop** (get property) steht für Anzeige des Wertes eines Merkmals. Es benötigt zwei Eingaben: ein Objekt und einen Merkmal-Namen.
- plist** (property list) steht für Ausgabe der vollständigen Merkmalliste des Wortes. Es benötigt eine Eingabe: ein Objekt.

```
?pprop "Huck.Finn" "Autor" "Twain"
?pprop "Huck.Finn" "Art" "Roman"
?pprop "Huck.Finn" "Seiten" "215"
?gprop "Huck.Finn" "Autor"
Twain
?plist "Huck.Finn"
[Seiten 215 Art Roman Autor Twain]
```

## 9.5 Erweitern Sie das Telefonverzeichnis unter Verwendung von Merkmallisten

Im nachfolgenden Beispiel wird gezeigt, wie Sie Ihr persönliches Telefonverzeichnis, das Sie zu Beginn dieser Lektion erstellt haben, erweitern können. (Die Namen sind lediglich Beispiele - Sie können dafür Ihre eigenen einsetzen.)

### Übung 9-4: Ein persönliches Telefonbuch erstellen

Ihre Angaben	Erläuterung
<code>?pprop "Egon" "Tel" 518965</code>	Ordnen Sie Egon das Merkmal <code>Tel</code> zu, und fügen Sie den Wert <code>518965</code> an.
<code>?gprop "Egon" "Tel" 518965</code>	Testen Sie <code>pprop</code> unter Verwendung von <code>gprop</code> . Bringen Sie den Wert des Merkmals <code>Tel</code> für Egon zur Anzeige.
<code>?pprop "Egon" "Geburtstag" [9 11]</code>	Ordnen Sie das Merkmal <code>Geburtstag</code> Egon zu, und fügen Sie den Wert <code>[9 11]</code> (= 11. September) an.
<code>?pprop "Egon" "Schule" "Drais"</code>	Ordnen Sie Egon den Wert <code>Schule</code> zu, und fügen Sie den Wert <code>Drais</code> an.
<code>?plist "Egon" [Schule Drais Geburtstag [9 11] Tel 518965]</code>	Bringen Sie die Liste aller Merkmale und der zugehörigen Werte, die Egon zugeordnet sind, zur Anzeige.
<code>?gprop "Egon" "Schuhgrösse" [ ]</code>	Für <code>Schuhgrösse</code> gibt es in der Merkmalliste von Egon keinen Merkmal-Namen. Wenn Dr. Logo keinen Merkmal-Namen finden kann, wird eine leere Liste <code>[ ]</code> angezeigt.

## 9.6 Zusammenfassung

Was Sie Neues gelernt haben:

- o Jedem Wort im Wortschatz von Dr. Logo ist eine Merkmalliste zugeordnet, in der eine Art Beschreibung enthalten ist und woraus diese besteht. Beispiel: Die Beschreibung könnte *Noten* und deren Wert *gut* sein.

## Neue Grundwörter:

Name	Eingabe
"	1 Wort
list	2
make	2 (Name und Definition)
thing oder :	1 Name
pprop	3 (Objekt, Merkmal und Wert)
gprop	2 (Objekt und Merkmal)
plist	1 Objekt
pons	keine

## 10 Wortspiele

Sie haben bereits gelernt, wie die Listen-Grundwörter `first`, `last`, `bf` und `b1` bei Listen mit Zahlen arbeiten. Jetzt, da Sie auch wissen, wie Dr. Logo Wörterlisten anlegt, können Sie diese Grundwörter und einige weitere für Wortspiele verwenden. Im folgenden Kapitel wird erklärt, wie man das macht.

### Lernziele

1. Sie sollen lernen, wie Sie Sätze zur Anzeige bringen können.
2. Sie sollen ein Spiel erstellen, das lustige Geschichten anzeigt.
3. Sie sollen lernen, Prozeduren zu schreiben, die eine Reaktion von Ihnen erwarten, und darauf wiederum antworten.

### 10.1 Wie Sätze zur Anzeige gebracht werden

Dr. Logo kennt noch mehr Grundwörter als diejenigen, die Sie für das Ändern von Listen bereits gelernt haben.

`lput` (last put) benötigt zwei Eingaben: ein einzelnes Objekt und eine Liste. Es setzt das Objekt als letztes Element in die Liste.

`fput` (first put) benötigt ebenfalls zwei Eingaben: ein einzelnes Objekt und eine Liste. Es setzt das Objekt als erstes Element in die Liste.

```
?lput 6 [2 4]
[2 4 6]
?fput 3 [5 7 9]
[3 5 7 9]
?lput "Kirschen [Jeder Junge ißt gern]
[Jeder Junge ißt gern Kirschen ]
```

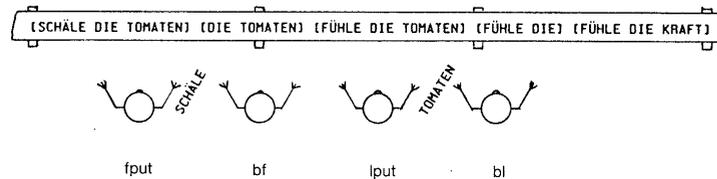
In einem der Beispiele zum Erlernen der Benennung haben Sie ein Kennwort wie folgt benannt:

```
?make "kennwort [Fühle die Kraft]
```

Wenn Sie jetzt verschiedene Grundwörter verwenden, können Sie erneut sehen, wie Dr. Logo ein *Montageband*, bestehend aus diesen Prozeduren, verwendet, um ein fertiges Produkt zu liefern.

```
?lput "Tomaten bl :kennwort
[Fühle die Tomaten]
?fput "Schäle bf lput "Tomaten bf :kennwort
[Schäle die Tomaten]
```

Um diese Zeilen auseinanderzuhalten, müssen Sie von rechts nach links lesen.



Im ersten Beispiel nimmt Dr. Logo die Definition von kennwort, d.h. [Fühle die Kraft].

- o Zunächst wird bl aus der Liste genommen, d.h. [Fühle die].
- o Dann wird das Element Tomaten als letztes Element in die Liste gesetzt, so daß [Fühle die Tomaten] entsteht.

Im zweiten Beispiel werden zwei weitere Schritte ausgeführt:

- o bf von [Fühle die Tomaten] wird genommen, d.h. [die Tomaten].
- o Das Element Schäle wird als erstes Element in die Liste gesetzt, so daß [Schäle die Tomaten] entsteht.

**Anmerkung:** lput und fput ändern den Wert von kennwort nicht.

## 10.2 Ein Spiel erstellen, das lustige Geschichten anzeigt

Im nachfolgenden Beispiel wird gezeigt, wie Sie eine Prozedur schreiben, um lustige Sätze aus berühmten Zeilen zusammenzustellen.

### Übung 10-1: Logo-Späße

Ihre Angaben

Erläuterung

Experimentieren Sie zunächst mit der Prozedur, bevor Sie sie benennen. Verwenden Sie den Reim: *Gretele, Pastetele, was machen Deine Gäns'? Sie sitzen im Wasser und waschen sich die Schwänz'!*

```
?fput "Anettele, bf [Gretele,
Pastetele]
Anettele, Pastetele
```

Dr. Logo nimmt bf der Liste [Pastetele] und setzt Anettele als erstes Element.

In der letzten Zeile muß das letzte Wort ersetzt werden.

```
?lput "Füss'! bl [und waschen!
sich die Schwänz'!]
[und waschen sich die Füss'!]
```

Dr. Logo nimmt bl der Liste [und waschen sich die] und setzt Füss'! als letztes Element.

```
?to spass :name :hauptwort
>pr
>fput :name bf [Gretele, Pastetele]
>pr lput :hauptwort bl [und waschen sich die Schwänz'!]
>end
spass defined
```

Schreiben Sie jetzt eine Prozedur, die zwei Eingaben, nämlich einen Namen und ein Objekt benötigt. Sie müssen Dr. Logo anweisen, was er mit jeder Zeile tun muß; fügen Sie daher das Grundwort pr ebenfalls ein.

```
?spass "Peterle, "Flügel!
Peterle, Pastetele,
was machen deine Gäns'?
Sie sitzen im Wasser
and waschen sich die Flügel!
```

Probieren Sie jetzt Ihre Prozedur aus.

?spass "Danielale, "Köpf'!  
 Danielale, Pastetele,  
 was machen deine Gäns'?  
 Sie sitzen im Wasser  
 and waschen sich die Köpf'!

Sie können zwei weitere Grundwörter für Listen verwenden, se und shuffle.

se benötigt zwei Listen als Eingabe. Die beiden Listen werden in eine Liste zusammengelegt.

?se [Regnet's im April,] [macht der Mai, was er will]  
 [Regnet's im April, macht der Mai, was er will]

shuffle benötigt eine Liste als Eingabe. Es mischt die Elemente der Liste willkürlich.

?shuffle [3 4 5 6 7 ]  
 [5 7 4 6 3]  
 ?shuffle [wo bist du jetzt]  
 [bist wo jetzt du]  
 ?first shuffle [wo bist du jetzt]  
 du

Probieren Sie die nachfolgende Übung aus und schreiben unter Verwendung von se und shuffle eine Prozedur, die zusammengewürfelte Rezepte ausgibt.

### Übung 10-2: Ein lustiges Kochbuch

Ihre Angaben

```
?rezept
>pr se
> [Man nehme] first shuffle
> [[3 Eier] [einen Apfel]
> [ein Pfund Knoblauch]]
>pr first shuffle
> [[fein zerkleinern]
> [in Limonade auflösen]
> [3 Monate lang in Erde ein-
graben]
> [bei 300 Grad backen]]
>pr se
> [bestreuen mit]
> first shuffle [Petersilie !
Schnee Muskat]
>pr se
> [als Beilage]
> first shuffle
> [[Eis] [Schokoladekuchen]
> [Spinat] [Bratkartoffeln]]
>end
rezept defined
?
```

Erläuterung

Diese Prozedur benötigt keine Eingaben. Dr. Logo führt sie aus, sobald Sie rezept eintippen. Achten Sie darauf, wie die eckigen Klammern verwendet werden, um die Elemente nach shuffle in eine Liste zu bringen.

```
?rezept
Man nehme 3 Eier
fein zerkleinern
bestreuen mit Schnee
als Beilage Bratkartoffeln
```

So, probieren Sie jetzt die Prozedur rezept. Es gibt 144 verschiedene Kombinationen!

```
?rezept
Man nehme ein Pfund Knoblauch
backen bei 300 Grad
bestreuen mit Petersilie
als Beilage Eis
```

```
?rezept
Man nehme ein Pfund Knoblauch
auflösen in Limonade
bestreuen mit Muskat
als Beilage Bratkartoffeln
```

### 10.3 Prozeduren, die eine Reaktion von Ihnen erwarten und darauf wiederum antworten

Dr. Logo kann Fragen stellen, auf Ihre Antwort warten und entsprechend Ihrer Reaktion verschiedene Prozeduren ausführen.

Wenn Sie Prozeduren mit Eingaben ausführen, hängt das Ergebnis der Prozedur von den von Ihnen gemachten Eingaben ab. Stellen Sie sich jetzt ein Programm vor, das bei der Ausführung Fragen stellt und entsprechend Ihrer Antworten verschieden reagiert. Sie können in Ihren Prozeduren unter Verwendung des Grundwortes `r1` Fragen einfügen.

`r1` (read list) wartet darauf, daß Sie eine Zeile eingeben, und gibt dann diese Zeile als Liste aus.

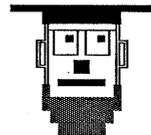
```
?r1
Guten Tag,
[Guten Tag,]
```

Nachfolgend ein Beispiel, wie Sie dieses Grundwort in einer Prozedur verwenden können:

```
?to grüsse
>pr [Wie heisst du?]
>pr se [Guten Tag,] r1
>end
grüsse defined
?grüsse
Wie heisst du?
Daniel
Guten Tag, Daniel
?
```

Wenn Sie die Grundwörter `if` und `r1` zusammen benutzen, können Sie eine Prozedur schreiben, die je nach Antwort unterschiedlich reagiert.

```
?to weltall
>pr [Wer war der erste Mann auf dem Mond?]
>pr [(Vor- und Nachname)]
>if r1 = [Neil Armstrong]
> [pr [Das ist richtig!] stop]
>pr [Versuche es noch einmal]
>weltall
>end
weltall defined
?weltall
Wer war der erste Mann auf dem Mond?
(Vor- und Nachname)
John Glenn
Versuche es noch einmal
Wer war der erste Mann auf dem Mond?
(Vor- und Nachname)
Neil Armstrong
Das ist richtig!
```



### Dr. Logo's Leitfaden

`r1` setzt die Antwort immer in eine Liste. Wenn Sie eine Prozedur schreiben, die `if` zum Vergleich der Antworten verwendet, dann vergewissern Sie sich, daß Sie eckige Klammern um die Antwort gesetzt haben. Dr. Logo beurteilt [Neil Armstrong] und Neil Armstrong unterschiedlich.

Im Dr-Logo-Lexikon wird erklärt, wie Sie das Grundwort `uc` dazu verwenden, um aus [Neil Armstrong] eine richtige Antwort zu bilden.

Das folgende Beispiel zeigt, wie Sie die Prozedur `weltall` als Baustein in einer Prozedur verwenden können, die verschiedene Fragen über das Raumfahrtprogramm stellt.

## Übung 10-3: Eine Prozedur schreiben, die Rückfragen stellt

## Ihre Angaben

```
?to weltraum
>weltall
>weltall1
>weltall2
>end
weltraum defined
```

```
?to weltall1
>pr [Wie heisst die amerikani-
sche Raumfahrtstation?]
>if r1 = [Skylab]
> [pr [Das ist richtig!] stop]
>pr [Versuche es noch einmal]
>weltall1
>end
weltall1 defined
```

```
?to weltall2
>pr [Welche amerikanische Fra-
u war als erste im Weltraum?]
>pr [(Vor- und Nachname)]
>if r1 = [Sally Ride]
> [pr [Das ist richtig!] stop]
>pr [Versuche es noch einmal]
>weltall2
>end
weltall2 defined
```

```
?weltraum
Wer war der erste Mann auf dem Mond?
(Vor- und Nachname)
Neil Armstrong
Das ist richtig!
Wie heisst die amerikanische Raumfahrts-
tation?
Spacelab
Versuche es noch einmal
Skylab
Das ist richtig!
```

## Erläuterung

Wenn Sie dieses Programm schreiben, verwenden Sie das "Von-oben-nach-unten"-Verfahren; d.h. schreiben Sie zuerst das fertige Programm, und definieren Sie dann die einzelnen Bausteine!

Welche amerikanische Frau war als erste!  
im Weltraum? (Vor- und Nachname)

Sally Ride  
Das ist richtig!  
?

Probieren Sie jetzt die Prozedur aus.

In dieser Übung taucht das Wort `r1` nur einmal in jeder Prozedur auf. Wenn die eingetippte Zeile der Antwort in der Prozedur entspricht, dann antwortet die Prozedur mit `Das ist richtig!` Wenn die eingetippte Zeile irgend etwas anderes enthält, antwortet die Prozedur mit `Versuche es noch einmal`. Diese Prozedur enthält nur eine Bedingung.

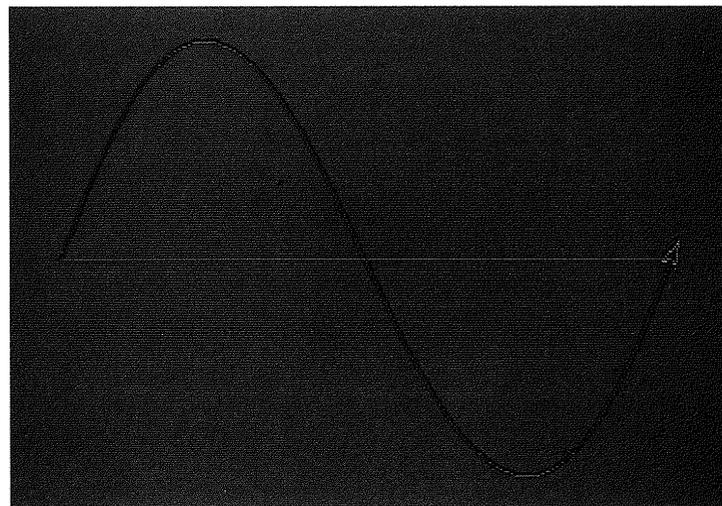
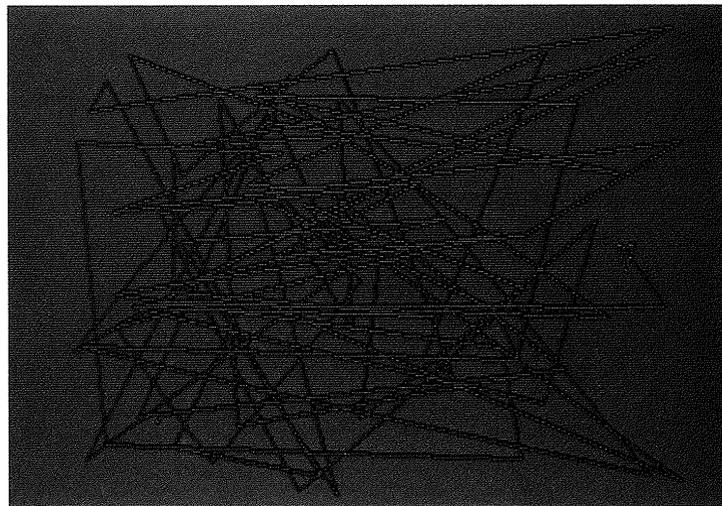
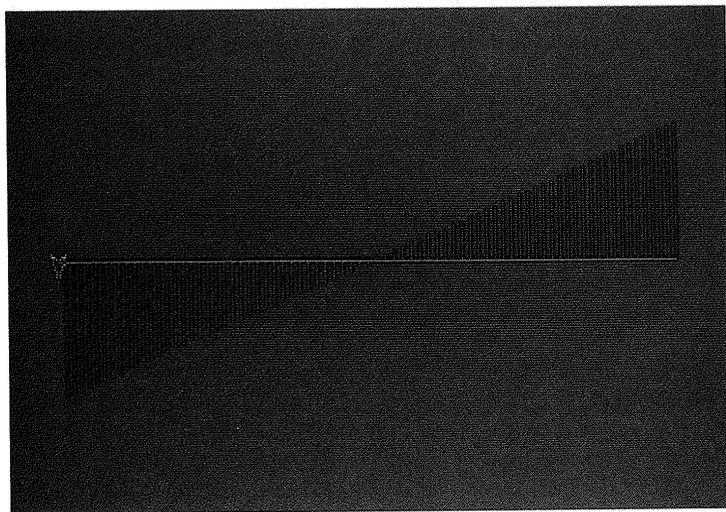
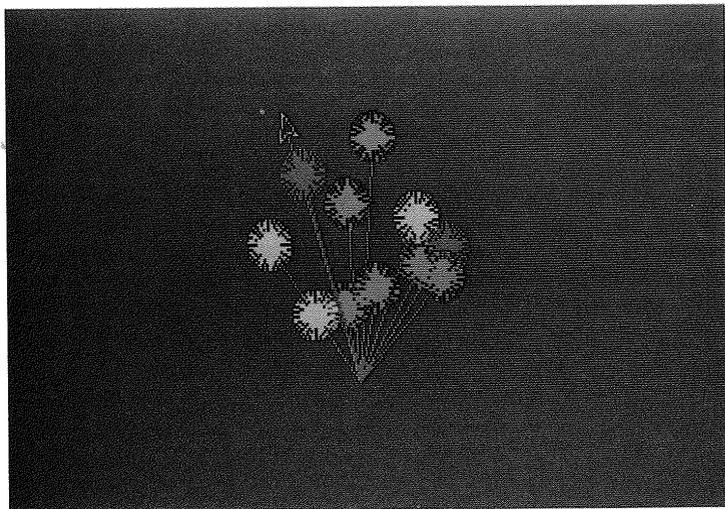
Wenn man mit `r1` mehrere Bedingungen stellen möchte, dann muß man sich folgende Eigentümlichkeit klarmachen: Die Prozedur wartet jedesmal, wenn `r1` erscheint, auf eine Antwort. Sehen Sie sich das nächste Beispiel an:

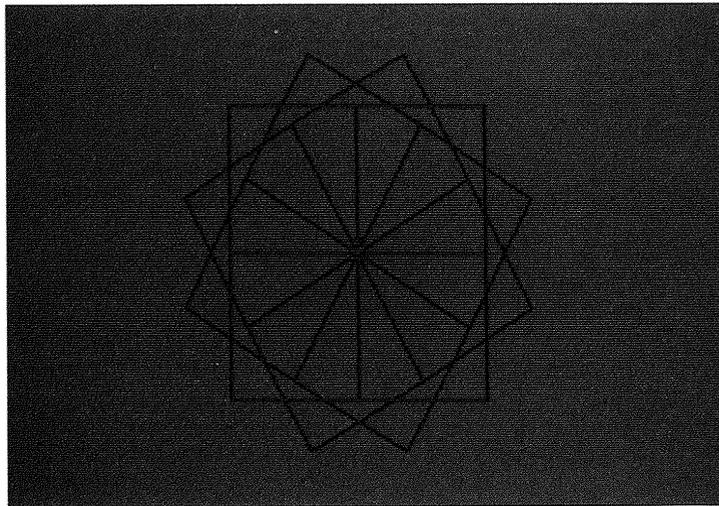
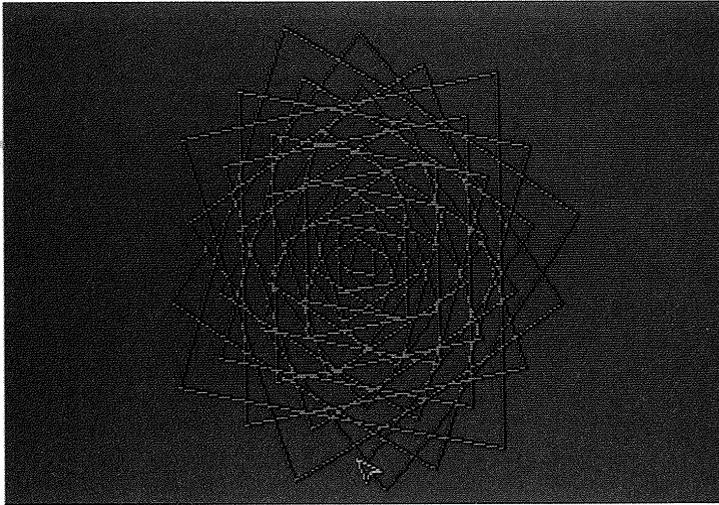
```
?to Familie
>pr [Wie heisst du?]
>if r1 = [Carola]
> [pr [Guten Tag, Mami] stop]
>if r1 = [Werner]
> [pr [Guten Tag, Papi] stop]
>if r1 = [Bianca]
> [pr [Guten Tag, kleine Schwester] stop]
>end
Familie defined
```

Wenn `r1` `[Carola]`, d.h. die erste Bedingung erfüllt ist, arbeitet die Prozedur problemlos; Dr. Logo folgt den in eckigen Klammern stehenden Anweisungen und beendet die Ausführung. Ansonsten stoppt Dr. Logo an der nächsten Zeile und wartet auf eine weitere Antwort, da das Wort `r1` erneut erscheint. Probieren Sie die Prozedur aus, um dies zu sehen.

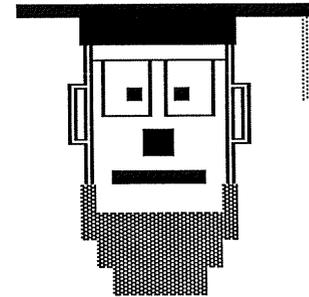
Die Lösung besteht darin, die Prozedur zu editieren, so daß `r1` nur einmal in der Prozedurdefinition auftaucht. Eine Möglichkeit stellt die Verwendung der Benennung (vorhergehende Lektion) dar, um aus `r1` irgend etwas anderes, wie z.B. das Folgende, zu machen:

```
?make "wahl r1
Truthahn
?:wahl
[Truthahn]
?
```





**Dr. Logo**



**Lexikon**

## Inhaltsverzeichnis

<b>1</b>	<b>Starten des Systems</b>	<b>11</b>
1.1	Starten von Dr. Logo	11
1.2	Definieren von Dr.-Logo-Prozeduren	12
1.3	Abspeichern der Dr.-Logo-Dateien	14
1.4	Laden von Dr.-Logo-Dateien	15
1.5	Festlegung des Standard-Laufwerkes	16
1.6	Erstellen einer Sicherungskopie einer Diskette	17
1.7	Beendigung der Sitzung	17
<b>2</b>	<b>Komponenten von Dr. Logo</b>	<b>19</b>
2.1	Der Zeichensatz von Dr. Logo	20
2.2	Objekte	23
2.2.1	Worte	23
2.2.2	Listen	25
2.2.3	Zahlen	27
2.2.4	Variablen	29
2.3	Zeilen und Ausdrücke	30
<b>3</b>	<b>Editier-Kommandos</b>	<b>33</b>
3.1	Zeileneditieren	33
3.2	Prozedureditieren	38
3.3	Texteditieren	39
3.4	Weitere Kontrollzeichenkommandos und Funktionstasten	45
<b>4</b>	<b>Arbeiten mit Prozeduren</b>	<b>49</b>
4.1	Benennung und Definition der Prozeduren	49
4.2	Schreiben übersichtlicher Prozeduren	50
4.2.1	Einrückungen	51
4.2.2	Kommentare	51
4.3	Verwendung von Variablen in Prozeduren	52
4.3.1	Definieren der Eingabeparameter	53
4.3.2	Informationsübergabe zwischen Prozeduren	54
4.4	Variablenarten	55
4.4.1	Erzeugen globaler Variablen	56
4.4.2	Erzeugen lokaler Variablen	57
4.5	Klassifikation der Prozeduren	60
4.6	Auswertung der Prozeduren	61

<b>5</b>	<b>Textbildschirm und Grafikbildschirm</b>	<b>63</b>
5.1	Bildschirme, Zeichenflächen und Fenster	63
5.2	Text- und Graphikausgabe	63
5.3	Bildschirmauflösung	64
5.4	Das Textfenster	64
5.4.1	Textkoordinaten auf geteiltem Bildschirm	65
5.5	Die Zeichenfläche	66
5.5.1	Die Koordinaten des Igels	66
5.5.2	Textkoordinaten auf der graphischen Zeichenfläche	68
5.5.3	Graphische Farben	68
5.6	Bildschirmattribute	68
5.6.1	Attribute der graphischen Zeichenfläche	69
5.6.2	Attribute des Igels	69
<b>6</b>	<b>Merkmallisten und Speicherverwaltung</b>	<b>71</b>
6.1	Merkmallisten	71
6.2	Verwaltung des Arbeitsspeicherbereiches	73
6.2.1	Die Größe des Arbeitsspeicherbereiches	73
6.2.2	Speichersäuberung	74
<b>7</b>	<b>Disketten, Laufwerke und Dateien</b>	<b>75</b>
7.1	Laufwerke	75
7.2	Disketten	76
7.3	Dateien	76
7.4	Dateibezeichnung	76
7.4.1	Was enthält eine Dateibezeichnung?	76
7.4.2	Bezugnahme auf mehrere Dateien	77
7.4.3	Dr. Logos Dateitypen	78
7.5	Logo-Dateien	79
7.5.1	Anwendung von save zur Erzeugung einer Logo-Datei	79
7.5.2	Anwendung von edf zur Erzeugung einer Logo-Datei	80
7.5.3	Bilddateien	80
<b>8</b>	<b>Wörterbuch</b>	<b>83</b>
8.1	Format der Kernbefehle	83
8.2	Mengensymbole	83
8.3	Typensymbole	84
8.4	Die Kernbefehle	85
	and	86
	arctan	86
	ascii	86
	bf	87
	bk	87

bl	88
buttonp	89
bye	89
catch	90
changeF	91
char	91
clean	92
copyoff	92
copyon	93
cos	93
cs	93
co	94
count	95
ct	95
cursor	95
defaultd	96
define	96
dir	97
dirpic	97
dot	97
dotc	98
ed	99
edall	100
edf	100
emptyt	101
end	102
ent	102
env	103
er	103
erall	104
erasefile	104
erasepic	105
ern	105
error	105
fd	107
fence	108
fill	108
first	108
fput	109
fs	109
glist	110
go	110
gprop	111

home	112
ht	113
if	114
int	115
item	115
keyp	116
label	117
last	117
lc	118
list	118
listp	118
load	118
loadpic	119
local	119
lput	119
lt	120
make	121
memberp	121
namep	122
nodes	122
noformat	122
not	123
notrace	123
nowatch	124
numberp	124
op	125
or	125
paddle	127
pal	127
pause	128
pd	128
pe	129
piece	130
plist	130
po	131
poall	131
pons	131
pops	131
pots	132
pprop	132
pps	133
pr	133
* (Produkt)	133

pu	134
px	135
quotient	136
random	137
rc	137
recycle	138
release	138
remainder	139
remprop	139
repeat	140
rerandom	140
rl	141
round	141
rq	142
rt	142
run	142
save	144
savepic	144
se	145
setbg	145
setcursor	146
setd	146
seth	147
setpal	147
setpc	148
setpos	148
setscrunch	148
setsplit	149
setx	149
sety	149
sf	150
show	150
shuffle	151
sin	151
sound	151
ss	153
st	153
stop	154
+ (Summe)	155
text	156
tf	156
thing	157
throw	157

to	158	
towards	159	
trace	160	
ts	161	
type	161	
uc	162	
wait	163	
watch	164	
where	165	
window	165	
word	165	
wordp	166	
wrap	166	
<b>9</b>	<b>Spezialwörterbuch</b>	<b>167</b>
	<b>Anhänge</b>	
A	Kontrollzeichenkommandos	181
B	DR Logos Systemkernbefehle	188
C	DR Logos Systemvariablen	189
D	DR Logos Systemmerkmale	190
E	DR Logos Fehlermeldungen	191
F	Kommandoliste	193
G	Notenliste	200
H	Rechnerspezifische Daten	203
	<b>Stichwortverzeichnis</b>	<b>204</b>
	Übersicht weiterer Markt&Technik-Produkte	210

# 1 Starten des Systems

Dieses Kapitel führt Sie durch eine komplette, abgeschlossene Sitzung mit Dr. Logo. Es wird gezeigt, wie Sie Dr. Logo starten und ausschalten, eine Prozedur definieren, Dateien abspeichern und laden, das aktuelle Laufwerk angeben und Ihre Dateien kopieren können. Sie brauchen eine formatierte Diskette, auf der Sie das Abspeichern und Kopieren Ihrer Dr.-Logo-Dateien üben können. Wenn Sie keine formatierte Diskette bereit haben, verwenden Sie das Dienstprogramm DISCKIT3 (CPC6128) bzw. DISCKIT (Joyce), um eine Diskette zu formatieren, bevor Sie mit der Arbeit beginnen.

In Kapitel 2 wird die Grammatik der Dr.-Logo-Sprache erklärt, d.h. warum und wann Sie Anführungszeichen und andere Satzzeichen eingeben müssen. Um diese Einführung kurz zu halten, werden diese Erläuterungen hier nicht wiederholt. Um diese Sitzung durchzuführen, geben Sie jedes der Beispiele genauso in Ihren Computer ein, wie es im Buch steht, und drücken Sie danach die Return-Taste. Die Antworten von Dr. Logo heben sich durch eine feinere Schrift ab. Machen Sie sich keine Sorgen, wenn Sie einen Tippfehler machen; durch Tippfehler können Sie weder Dr. Logo noch Ihrem Computer Schaden zufügen.

Wenn Sie in diesem Kapitel auf einen für Sie unverständlichen Ausdruck aus der Computer-Fachsprache stoßen, können Sie im *Spezialwörterbuch* nachschlagen.

## 1.1 Starten von Dr. Logo

1. Schalten Sie den Computer samt Bildschirm und - bei Bedarf - den Drucker ein.
2. Laden Sie das Betriebssystem: beim CPC6128 durch Einlegen der ersten Seite der CP/M-Plus-Diskette und Eintippen von `!cpm` bzw. beim Joyce allein durch Einlegen der zweiten Seite der CP/M-Plus-Diskette (oder einer Kopie der Systemdiskette).
3. Nehmen Sie die Systemdiskette aus dem Laufwerk und ersetzen Sie sie durch die Arbeitskopie der Dr.-Logo-Diskette. Wenn Ihr Computer nur ein Laufwerk besitzt, speichern Sie Ihre Programme entweder auf der Arbeitsdiskette, oder Sie nehmen diese heraus und legen eine leere formatierte Datendiskette ein. Wenn Ihr Computer zwei Laufwerke besitzt, ist es vielleicht einfacher, Sie verwenden das zweite Laufwerk für die neuen Daten.

Wenn A> auf Ihrem Bildschirm erscheint, geben Sie beim CPC6128 `submit logo3` bzw. beim Joyce `submit logo` ein. Beobachten Sie die rote Lampe am Laufwerk. Sie

soll blinken, und Sie müssen hören, wie Ihr System Dr. Logo von der Diskette in den Hauptspeicher des Computers lädt. Während Dr. Logo geladen wird, erscheint folgende Meldung auf dem Bildschirm:

```

Welcome to
Amstrad Logo, Version VX.X
Copyright (c) 1983, Digital Research, Inc.
Pacific Grove, California

```

```

Dr. Logo is trademark of
Digital Research
Product No. XXXX-0000

```

```
Please wait...
```

Die Versionsnummer VX.X zeigt Ihnen den Revisionsstand der Dr.-Logo-Version, die Sie besitzen. Nachdem diese Begrüßung verschwindet, erscheinen ein Fragezeichen und ein helles Rechteck, der Cursor (■), auf Ihrem Bildschirm. Das Fragezeichen zeigt Ihnen, daß Dr. Logo darauf wartet, daß Sie etwas über die Tastatur eingeben. Der Cursor zeigt, wo das nächste Zeichen erscheint, das Sie eingeben. Das Fragezeichen wird Prompt genannt, es zeigt, daß Sie sich in der Hauptebene befinden.

Nachdem Dr. Logo in den Hauptspeicher geladen wurde, teilt er Ihnen einen Teil des Hauptspeichers als Arbeitsspeicherbereich zu. Wenn Sie Dr. Logo starten, ist dieser Arbeitsspeicherbereich leer und steht zu Ihrer Verfügung, damit Sie neue Prozeduren eingeben oder früher abgespeicherte Prozeduren von einer Diskette laden können.

## 1.2 Definieren von Dr.-Logo-Prozeduren

Eine Prozedur ist eine Liste von Anweisungen, die Dr. Logo angibt, wie er einen bestimmten Auftrag durchführen soll. Dr. Logo führt diesen Auftrag durch, wenn Sie den Namen der Prozedur eingeben und die Return-Taste drücken. Die Liste der Anweisungen ist die Definition der Prozedur. Um eine Prozedur zu definieren, verwenden Sie Dr.-Logo-Kommandos, genannt Kernbefehle (bzw. Grundworte oder auch Primitive) und früher definierte andere Prozeduren. Hier ist ein Beispiel, das zeigt, wie man eine Prozedur eingeben kann, die Dr. Logo mit dem Igel ein Quadrat zeichnen läßt:

```

?to quadrat
>repeat 4 [fd 50 rt 90]
>end
quadrat defined
?

```

Wenn Sie `to quadrat` eingeben, antwortet Dr. Logo mit einem neuen Prompt, dem Größerzeichen `>`. Das spezielle Wort `to` zeigt Dr. Logo, daß Sie eine neue Prozedur definieren wollen. Das Prompt `>` zeigt Ihnen, daß Dr. Logo Ihre jetzt einzugebenden Anweisungen als Definition einer Prozedur interpretieren wird. Wenn Sie die Eingabe Ihrer Anweisungen beendet haben, geben Sie `end` als Antwort auf das Prompt `>` ein. Dieses Wort zeigt Dr. Logo, daß Sie die Definition der Prozedur beendet haben.

Wenn Sie eine Prozedur definiert haben, können Sie Dr. Logo den neuen Auftrag durchführen lassen, indem Sie als Antwort auf sein Prompt `?` den Namen der Prozedur eingeben und die Return-Taste drücken. Geben Sie zum Beispiel folgendes ein:

```

?quadrat
?

```

Mit Hilfe des Befehls `cs` können Sie Ihre Graphik löschen:

```

?cs
?

```

Sie können Dr. Logo beibringen, wie er ein Dreieck zeichnen kann:

```

?to dreieck
>repeat 3 [fd 50 rt 120]
>end
dreieck defined
?dreieck
?

```

Jetzt können Sie die Prozeduren `dreieck` und `quadrat` verwenden, um eine andere Prozedur zu definieren. So zum Beispiel:

```

?cs
?to zeichne.haus
>quadrat
>fd 50
>dreieck
>end
zeichne.haus defined
?zeichne.haus
?

```

Sie können die Namen aller von Ihnen definierten Prozeduren mit Hilfe des Kernbefehls `pots` auflisten lassen:

```
?pots
to quadrat
to dreieck
to zeichne.haus
?
```

Wenn Sie Ihren Computer ausschalten, geht der Inhalt des Arbeitsspeichers verloren. Wenn Sie Ihren Computer zum Beispiel jetzt ausschalten würden, würden die oben definierten Prozeduren verlorengehen. Um eine Prozedur auch später verwenden zu können, müssen Sie sie auf Diskette abspeichern.

Es ist empfehlenswert, alle Ihre Dr.-Logo-Prozeduren auf einer Datendiskette, getrennt von Ihrer Dr.-Logo-Systemdiskette abzuspeichern. Eine Datendiskette ist eine leere Diskette, die mit Hilfe des Dienstprogramms DISCKIT3 bzw. DISCKIT formatiert wurde. Fertigen Sie Sicherungskopien sowohl von Ihrer Systemdiskette als auch von allen Ihren Datendisketten an.

### 1.3 Abspeichern der Dr.-Logo-Dateien

Abspeichern einer Datei bedeutet Kopieren des Inhalts Ihres Arbeitsspeicherbereichs zur permanenten Speicherung in eine Datei auf Diskette.

1. Wenn Sie einen Computer mit nur einem Laufwerk besitzen und nicht auf Ihrer Dr.-Logo-Systemdiskette speichern wollen, nehmen Sie diese aus dem Laufwerk heraus und ersetzen sie durch eine Datendiskette. Geben Sie das folgende Kommando ein, um den Inhalt des Arbeitsspeichers in eine Datei auf der Datendiskette abzuspeichern.

```
?save "haus
?
```

Wenn in Ihrem Computer zwei Laufwerke installiert sind oder wenn Sie die zweite Seite der Diskette zur Datenspeicherung nutzen wollen, stecken Sie die Datendiskette in das Laufwerk B: ein und geben Sie

```
?save "B:haus
```

ein. Damit haben Sie eine Kopie Ihrer drei Prozeduren (quadrat, dreieck und zeichne.haus) in einer Datei namens haus auf Ihrer Datendiskette angefertigt. Ihre Prozeduren sind im Arbeitsspeicher immer noch vorhanden, wie Sie durch Eingabe des Kernbefehls pots sehen können:

```
?pots
to quadrat
to dreieck
to zeichne.haus
?
```

2. Um sich davon zu überzeugen, daß Dr. Logo Ihre Prozeduren wirklich abgespeichert hat, bitten Sie ihn, die Namen der Dr.-Logo-Dateien auf der Diskette anzuzeigen:

```
?dir
[haus]
?
```

Oder, wenn Sie die Datei auf einer Diskette im Laufwerk B: bzw. auf Seite B: abgespeichert haben, geben Sie folgendes ein:

```
?dir "b:
[haus]
?
```

3. Jetzt, da Sie Ihre Prozeduren abgespeichert haben, können Sie sie aus dem Arbeitsspeicherbereich löschen; Sie können sicher sein, daß sie jederzeit abrufbar sind. Geben Sie folgendes ein:

```
?erall
?
```

Um sich davon zu überzeugen, daß Ihr Arbeitsspeicherbereich leer ist, lassen Sie Dr. Logo die Namen der Prozeduren im Arbeitsspeicher ausgeben:

```
?pots
?
```

Dr. Logo gibt nur das Fragezeichen aus, weil keine Prozedur im Arbeitsspeicher vorhanden ist.

### 1.4 Laden von Dr.-Logo-Dateien

Laden einer Datei bedeutet Kopieren der auf einer Diskette permanent abgespeicherten Prozeduren oder anderer Informationen in den Dr.-Logo-Arbeitsspeicher. Um eine Datei laden zu können, muß sie vorher auf eine Diskette abgespeichert wor-

den sein und Sie müssen den Namen dieser Datei mit dem `dir`-Kommando auflisten können.

- Um die soeben abgespeicherte Datei in Ihren Arbeitsspeicherbereich zu laden, geben Sie

```
?load "haus
?
```

ein, oder, wenn Sie die Datei auf der Diskette im Laufwerk B: bzw. Seite B: abgespeichert haben, geben Sie folgendes ein:

```
?load "b:haus
?
```

Sie können jetzt beobachten, wie Dr. Logo die Prozeduren von der Diskette in den Arbeitsspeicher lädt und folgende Liste auf dem Bildschirm ausgibt:

```
quadrat defined
dreieck defined
zeichne.haus defined
?
```

- Prüfen Sie jetzt, ob die Prozeduren im Arbeitsspeicher wirklich vorhanden sind. Diese Prüfung können Sie entweder durch Ausführen der Prozeduren oder durch Anwendung von `pots` zum Ausgeben ihrer Namen vornehmen:

```
?zeichne.haus

?pots
to quadrat
to dreieck
to zeichne.haus
?
```

## 1.5 Festlegung des Standard-Laufwerkes

Wenn Sie in einem Disketten-Kommando, wie zum Beispiel `save`, `load` oder `dir` kein Laufwerk angeben, werden die Dateien von Dr. Logo automatisch auf die Diskette im Standard-Laufwerk geschrieben, bzw. von dort gelesen. Wenn Sie Dr. Logo vom Prompt `A>` aus starten, ist `A:` das Standard-Laufwerk. Wenn Sie ein System mit nur einem Laufwerk besitzen, steht Ihnen nur Laufwerk `A:` zur Verfügung, und Sie brauchen die Laufwerksbezeichnung in keinem Kommando anzugeben. Wenn Sie

ein System mit zwei Laufwerken besitzen und Ihre Systemdiskette im Laufwerk `A:` belassen wollen, geben Sie folgendes zur Festlegung des Laufwerks `B:` als Standard-Laufwerk ein:

- Bestimmen Sie zuerst, welches Laufwerk jetzt als Standard-Laufwerk angegeben ist:

```
?defaultd
A:
?
```

- Legen Sie jetzt Laufwerk `B:` als Standard-Laufwerk fest:

```
?setd "b:
?defaultd
B:
?
```

## 1.6 Erstellen einer Sicherungskopie einer Diskette

Erstellen einer Sicherungskopie bedeutet Kopieren des vollen Inhalts einer Diskette auf eine andere Diskette. Professionelle Programmierer vermeiden Datenverlust durch Erstellen von Sicherungskopien von wertvollen Disketten und Dateien. Wenn Ihre Arbeitskopie zufällig beschädigt oder gelöscht wird, können Sie die entsprechende Datei von Ihrer Sicherungskopie wieder laden. Sie können mehrere Sicherheitskopien einer Datei herstellen, indem Sie den in Abschnitt 1.3 beschriebenen Vorgang des Abspeicherns mehrfach ausführen.

Um eine gesamte Diskette zu sichern, kopieren Sie sie mit den Hilfsprogrammen `DISCKIT3` bzw. `DISCKIT`.

Sie können auch von einer einzigen Datei eine Kopie erstellen: zu diesem Zweck verwenden Sie das `CP/M`-Dienstprogramm `PIP`. Bevor Sie eine Kopie einer Datei erstellen können, müssen Sie diese Datei mit `save` aus Ihrem Arbeitsspeicherbereich auf eine formatierte Datendiskette schreiben.

## 1.7 Beendigung der Sitzung

Bevor Sie Ihren Computer ausschalten, prüfen Sie, ob Sie folgende Schritte bereits ausgeführt haben:

- Speichern Sie den Inhalt Ihres Arbeitsspeicherbereichs mit `save` auf einer Diskette ab, damit Sie Ihre Dateien später mit `load` wieder laden können.

2. Fertigen Sie Sicherungskopien Ihrer Dateien auf einer anderen Diskette an.
3. Nehmen Sie Ihre Disketten aus den Laufwerken heraus. Sie dürfen Ihr System nie ausschalten, wenn sich noch Disketten in den Laufwerken befinden. Das Ausschalten kann Ihren Disketten bleibende Schäden zufügen.
4. Schalten Sie die Netzschalter Ihres Monitors bzw. Ihrer Systemeinheit und des Druckers auf Stellung *AUS* bzw. *OFF*.

## 2 Komponenten von Dr. Logo

Wenn Sie Ihr Dr.-Logo-System zum ersten Male starten, kennt es nahezu 200 Kommandos, Kernbefehle genannt. Diese Kernbefehle, wie zum Beispiel `pr` oder `fd`, stellen Operationen oder Prozeduren dar, die Dr. Logo ausführen kann, ohne sie zuerst lernen zu müssen. Der Name einer Prozedur zusammen mit einem beliebigen nachfolgenden Eingabeparameter, den die Prozedur erwartet, wie zum Beispiel `fd 50`, ist ein Ausdruck. Hier ist `50` der Eingabeparameter von `fd`, und `fd` und `50` zusammen sind ein Ausdruck. Dr. Logo kann je einen Ausdruck auswerten, den Sie als Antwort auf das Prompt `?` eingeben. So zum Beispiel

```
?pr "hallo
hallo
?2 + 2
4
?
```

Mit Hilfe der Kernbefehle können Sie auch Ihre eigenen Prozeduren definieren. Eine Prozedur zu definieren bedeutet Dr. Logo eine neue Operation beizubringen. So zum Beispiel

```
?to sage.hallo
>pr "hallo
>end
sage.hallo defined
?
```

Jetzt ist `sage.hallo` definiert, und Sie können `sage.hallo` wie einen Kernbefehl verwenden:

```
?sage.hallo
hallo
?
```

Wenn Sie Prozeduren erzeugt haben, können Sie sie abspeichern und später wieder verwenden. Sie können Ihre Prozeduren ändern, löschen oder Ihre persönliche Prozedurbibliothek mit anderen Prozeduren kombinieren, um komplexere Programme zu bilden. Die Prozeduren sind die grundlegenden Bausteine der Dr.-Logo-Programme.

Beachten Sie, daß ein Kernbefehl einfach eine Prozedur ist, die Dr. Logo bereits kennt. Alles, was Sie eingeben, ist entweder Prozedurname oder Eingabeparameter für irgendeine Prozedur.

Im obigen Beispiel ist `pr "hallo` ein Ausdruck. Der Prozedurname, `pr`, identifiziert einen Kernbefehl, der seinen Eingabeparameter, `hallo`, auf dem Bildschirm ausgibt. Wenn Sie eine Zeile in der Hauptebene (d.h. nach dem Prompt) eingeben, müssen Sie die Return-Taste drücken, um Dr. Logo zu sagen, daß er die Ausdrücke in der betreffenden Zeile auswerten soll.

Eine Prozedur kann eine bestimmte Anzahl und Art von Eingabeparametern verlangen. Als Eingabeparameter zu Prozeduren können Worte, Zahlen oder Listen dienen. In diesem Kapitel wird erklärt, wie man Prozedurnamen mit Eingabeparametern kombinieren kann, um von Dr. Logo auswertbare Ausdrücke zu erzeugen. Danach wird erklärt, wie Dr. Logo eine Zeile auswertet, in der Sie mehr als einen Ausdruck eingegeben haben.

## 2.1 Der Zeichensatz von Dr. Logo

Zum Erzeugen eines Prozedurnamens können Sie fast alle Zeichen auf der Tastatur verwenden, einschließlich Groß- und Kleinbuchstaben, Zahlen und Symbolen. Als erstes Zeichen in einem Prozedurnamen darf keine Zahl verwendet werden, ein Prozedurname darf auch nicht aus einer einzigen Zahl bestehen. Zum Beispiel:

Zulässig	Unzulässig
Kiste1	1steKiste
kiste2	2teKiste
VogelNest	
zeichne&g	

Bedenken Sie bei der Vergabe der Namen jedoch, daß Dr. Logo sich diese in genau der Schreibweise merkt. Die Prozedur `Quadrat` ist daher eine andere als die Prozedur `quadrat`.

In Dr. Logo haben einige Zeichen eine besondere Bedeutung. Dr. Logo erkennt einige Zeichen, wie zum Beispiel `"`, `[` und `]` sowie `:` als Satzzeichen für Worte, Listen und Variablen (siehe Abschnitte 2.2.1 und 2.2.4). Andere Zeichen, *Begrenzer* genannt, können ein Zeichen, ein Wort oder eine Zahl vom übrigen Text abheben. So ist zum Beispiel ein Leerzeichen ein Begrenzer.

Einige Zeichen, wie `+` und `-` sind eigentlich Kernbefehle. Einige aus nur einem Zeichen bestehende Kernbefehle werden gelegentlich *eingebettete* oder *vorangestellte* Kernbefehle genannt, weil Eingabeparameter zu einem Kernbefehl dieser Art in einer anderen Reihenfolge eingegeben werden können als zu einem voll ausgeschriebenen Kernbefehl. Nehmen wir folgendes Beispiel:

```
?pr "hallo
hallo
?5 + 4
9
?
```

Der Kernbefehl `pr` ist ein vorangestellter Kernbefehl, da er vor seinem Eingabeparameter stehen muß. Im zweiten Beispiel ist das Pluszeichen `+` ein eingebetteter Kernbefehl, da er zwischen zwei Eingabeparametern steht. Allerdings können Sie das Pluszeichen `+` auch als vorangestellten Kernbefehl verwenden:

```
?+ 5 4
9
?
```

Einige Zeichen sind sowohl Begrenzer als auch Satzzeichen oder Kernbefehle. In Tabelle 2-1 sind die Begrenzer von Dr. Logo aufgelistet.

Tabelle 2-1: Begrenzer von Dr. Logo

Zeichen	Typ	Funktion
[		Zeigt den Anfang einer Liste an
]		Zeigt das Ende einer Liste an
(		Zeigt den Anfang einer Ausdrucksgruppe an
)		Zeigt das Ende einer Ausdrucksgruppe an
;		Zeigt den Anfang eines Kommentars an
=	logisch	Vorangestellter oder eingebetteter Kernbefehl, der TRUE ausgibt, wenn das erste Wort und das zweite Wort gleich sind
<	logisch	Vorangestellter oder eingebetteter Kernbefehl, der TRUE ausgibt, wenn das erste Wort kleiner ist als das zweite
>	logisch	Vorangestellter oder eingebetteter Kernbefehl, der TRUE ausgibt, wenn das erste Wort größer ist als das zweite
+	arithmetisch	Vorangestellter oder eingebetteter Kernbefehl, der die Summe der Eingaben ausgibt

Zeichen	Typ	Funktion
-	arithmetisch	Vorangestellter oder eingebetteter Kernbefehl, der die Differenz der Eingaben ausgibt
*	arithmetisch	Vorangestellter oder eingebetteter Kernbefehl, der das Produkt der Eingaben ausgibt
/	arithmetisch	Vorangestellter oder eingebetteter Kernbefehl, der den Quotienten aus den Eingaben ausgibt

Wenn Sie einen Begrenzer in einem Ausdruck verwenden, müssen Sie ihm kein Leerzeichen voranstellen, um ihn vom Rest der Zeile abzuheben. Wenn eines dieser Zeichen irgendwo auftritt, nimmt Dr. Logo stets an, daß es den Anfang eines neuen Wortes oder einer neuen Zahl markiert, das oder die von allen anderen Zeichen in der Zeile getrennt werden soll. Diese Eigenschaft von Dr. Logo macht die Eingabe der Ausdrücke in vielen Fällen einfacher, so zum Beispiel:

```
?256+1000
1256
?
```

Wenn Sie aber diese Zeichen in anderen Situationen verwenden, wird die Sache komplizierter. Wenn Sie das Trennzeichen zum Beispiel als Bindestrich statt als Minuszeichen verwenden, werden Sie sehen, daß Dr. Logo unerwünschte Leerzeichen in den ausgegebenen Text einfügt:

```
?pr [Igel-Grafik von hoher Auflösung]
Igel - Grafik von hoher Auflösung
?
```

Um Dr. Logo zu sagen, daß er einen Begrenzer als Bestandteil eines Wortes interpretieren soll, müssen Sie diesem Zeichen einen umgekehrten Schrägstrich (\) bzw. beim Joyce das große ö voranstellen:

```
?pr [Igel\-Grafik von hoher Auflösung]
Igel-Grafik von hoher Auflösung
?
```

Andere Zeichen, *Kontrollzeichen* genannt, werden von Dr. Logo als Befehle erkannt und interpretiert. Kontrollzeichen können zum Editieren, Unterbrechen und Beenden einer Prozedur und zum Umschalten zwischen Volltext-, Vollgraphik- und geteiltem Bildschirm verwendet werden. Die Kontrollzeichen sind in Kapitel 3 *Editier-Kommandos* und in Kapitel 5 *Textbildschirm und Graphikbildschirm* beschrieben.

## 2.2 Objekte

Zur Durchführung ihres Auftrags kann eine Prozedur eine bestimmte Anzahl von Eingabeparametern unterschiedlicher Art verlangen. Die verlangte Eingabe kann aus einem Wort oder mehreren Worten, Listen oder Zahlen bestehen. Wenn eine Prozedur Eingabeparameter verlangt, dann sagen wir, daß sie ein Objekt akzeptieren kann; und das bedeutet, daß die Prozedur zur Ausführung ihrer Tätigkeit ein Wort, eine Liste oder eine Zahl verlangt. Um Ihre Prozeduren flexibler zu gestalten, können Sie Objekte durch Namen, Variablen genannt, vertreten lassen. In diesem Kapitel werden Worte, Listen, Zahlen und Variablen beschrieben.

### 2.2.1 Worte

Ein Wort ist eine Gruppe von unmittelbar aufeinanderfolgenden Zeichen, die von den anderen Zeichen in der Zeile durch einen Begrenzer, wie zum Beispiel ein Leerzeichen, getrennt sind. Sie können Punkte oder Unterstreichungszeichen als Bindeglieder verwenden, um Dr. Logo mehrere Worte als ein Wort interpretieren zu lassen. Zum Beispiel:

```
?pr "Tafel.Schokolade
Tafel.Schokolade
?pr "Griechische_Rosinen
Griechische_Rosinen
?
```

Dr. Logo braucht etwas, um Eingabenamen von Prozedurnamen zu unterscheiden. Wenn Sie einen Ausdruck eingeben, müssen Sie dem Eingabenamen daher ein Anführungszeichen (") voranstellen. Wenn Sie einem Eingabenamen kein Anführungszeichen voranstellen, wird sich Dr. Logo beschweren, so zum Beispiel:

```
?pr "think
think
?pr think
I don't know how to think
?
```

Obwohl Anführungszeichen am Anfang eines Wortes die oben genannte Bedeutung haben, sind sie keine echte Begrenzer und können daher im Inneren eines Wortes auch als gewöhnliche Zeichen verwendet werden. So zum Beispiel:

```
?pr "Sage"hallo"hier
Sage"hallo"hier
?
```

Verwenden Sie keine Anführungszeichen am Anfang eines Wortes, es sei denn, Sie wollen damit einen Eingabennamen markieren. Wenn Sie Anführungszeichen am Anfang eines Eingabewortes haben wollen, verwenden Sie zwei Anführungszeichen:

```
?pr ""Sage_hallo"
"Sage_hallo"
?
```

Andere Kernbefehle können Worte Zeichen für Zeichen untersuchen, ein Wort abtrennen und ein Wort zusammenstellen. Wenn Dr. Logo ein Wort auf diese Weise manipuliert, betrachtet er jedes Zeichen im Wort als ein aus einem Zeichen bestehendes Wort. Zum Beispiel:

```
?first "Zebra
Z
?first first [zebra elefant kanguruh]
z
?bf "Atypisch
typisch
?last "grob
b
?bl "Daten
Date
?word "R2 "D2
R2D2
?
```

Wie bereits in Kapitel 2.1 erklärt wurde, können Sie Dr. Logo sagen, daß Sie ein Begrenzerzeichen in einem Wort als gewöhnliches Zeichen verwenden wollen, indem Sie dem Begrenzerzeichen einen umgekehrten Schrägstrich voranstellen. Ist allerdings das Begrenzerzeichen das erste Zeichen im betreffenden Wort und ist diesem Wort ein Anführungszeichen vorangestellt, brauchen Sie in den meisten Fällen keinen umgekehrten Schrägstrich zwischen dem Anführungszeichen und dem Begrenzerzeichen einzugeben.

```
?"+mehr
+mehr
?ascii "*"
42
?
```

Leerzeichen und eckige Klammern, [ ], werden auch dann als Begrenzer interpretiert, wenn ihnen ein Anführungszeichen vorangestellt wird. Verwenden Sie einen umgekehrten Schrägstrich \ bzw. das ö nach dem Anführungszeichen, um Dr. Logo

zu sagen, daß er ein Leerzeichen oder eckige Klammern als gewöhnliche Zeichen betrachten soll.

```
?ascii "[
ascii doesn't like an empty word as in!
put
?ascii "\[
91
?
```

(Das Ausrufezeichen im obigen Beispiel bedeutet, daß Dr. Logo den rechten Rand des Monitors bei der 40-Zeichen-Darstellung erreicht hat und die Meldung in der nächsten Zeile fortsetzt.)

Der Kernbefehl `rq` erzeugt ein Wort, das alle Zeichen beinhaltet, die in einer Zeile eingegeben wurden. Wenn die Zeile durch Leerzeichen getrennte Worte enthält, setzt `rq` umgekehrte Schrägstriche vor die Leerzeichen, damit die Leerzeichen von Dr. Logo als gewöhnliche Zeichen interpretiert werden. Damit wird aus einer Zeile ein Wort gemacht. Zum Beispiel:

```
?pr first rq
Hallo Welt
H
?
```

Sie können die umgekehrten Schrägstriche sichtbar machen, indem Sie ein mit Hilfe von `rq` erzeugtes Argument in den Editor laden. (In Kapitel 3 wird erklärt, wie Sie ein Argument in den Editor laden können.)

## 2.2.2 Listen

Eine *literale Liste* ist eine Folge von Objekten zwischen eckigen Klammern. Mit Hilfe der Kernbefehle `list` und `se` können Sie eine Liste auch ohne eckige Klammern erzeugen. Jedes Element einer Liste kann ein Wort, eine Zahl+ oder eine andere Liste sein. Innerhalb der Liste werden die Argumente durch Leerzeichen voneinander getrennt. In einer literalen Liste betrachtet Dr. Logo jedes Argument als ein literales Argument, so daß Sie den Worten innerhalb einer literalen Liste keine Anführungszeichen voranzustellen brauchen. Hier sind einige zulässige literale Listen:

```
[Grüsse!]
[L M N]
[[Brot Butter] [Suppe Sandwich]]
[10 20 [22 25] 30]
```

Die folgenden Beispiele zeigen Worte innerhalb und außerhalb der Listen und wie man eine Liste durch Anwendung von `se` und `list` erzeugen kann:

```
?(list "L "M "N)
[L M N]
?se [Ich mag] "Hamburger
[Ich mag Hamburger]
?
```

Die Kernbefehle, die zur Manipulation von Wortelelementen verwendet werden, dienen auch zur Manipulation von Listenelementen:

```
?bf [nicht wirklich]
[wirklich]
?bl [nicht wirklich]
[nicht ]
?fput 5 [10 20 [22 25] 30]
[5 10 20 [22 25] 30]
?
```

Nach Ausführung des Ausdrucks `fput` enthält obige Liste 5 Elemente, von denen das vierte selbst eine Liste ist. Folgendes Beispiel zeigt, wie Listen innerhalb von Listen zu zählen sind:

```
?count [5 10 20 [22 25] 30]
5
?count item 4 [5 10 20 [22 25] 30]
2
?
```

In Tabelle 2-2 werden die vier Kernbefehle miteinander verglichen, die zwei Argumente als Eingabeparameter erwarten und ein Wort oder eine Liste ausgeben.

Tabelle 2-2: Vergleich der Kernbefehle der Listenbearbeitung

Kern-befehl	Eingabe 1	Eingabe 2	Ausgabe
<code>list</code>	"gelb	"gruen	[gelb gruen]
<code>se</code>	"gelb	"gruen	[gelb gruen]
<code>fput</code>	"gelb	"gruen	gelbgruen
<code>lput</code>	"gelb	"gruen	gruengelb

Kern-befehl	Eingabe 1	Eingabe 2	Ausgabe
<code>list</code>	"Luft	[ist blau]	[Luft [ist blau]]
<code>se</code>	"Luft	[ist blau]	[Luft ist blau]
<code>fput</code>	"Luft	[ist blau]	[Luft ist blau]
<code>lput</code>	"Luft	[ist blau]	[ist blau Luft]

<code>list</code>	[sage Hallo]	[zu mir]	[[sage Hallo] [zu mir]]
<code>se</code>	[sage Hallo]	[zu mir]	[sage Hallo zu mir]
<code>fput</code>	[sage Hallo]	[zu mir]	[[sage Hallo] zu mir]
<code>lput</code>	[sage Hallo]	[zu mir]	[zu mir [sage Hallo]]
<code>list</code>	[Hallo]	[]	[[Hallo] []]
<code>se</code>	[Hallo]	[]	[Hallo]
<code>fput</code>	[Hallo]	[]	[[Hallo]]
<code>lput</code>	[Hallo]	[]	[[Hallo]]

### 2.2.3 Zahlen

Eine Zahl besteht aus einer oder mehreren Ziffern, die von den anderen Zeichen in der Zeile durch Leerzeichen oder andere Begrenzer getrennt sind. Eine Zahl ist etwas Ähnliches wie ein Wort, aber Sie brauchen einer Zahl kein Anführungszeichen voranzustellen, es sei denn, Sie wollen sie als Variablenamen verwenden. Sie wissen ja: Dr. Logo läßt nicht zu, einen Prozedurnamen mit einer Ziffer als erstes Zeichen zu definieren.

Im allgemeinen akzeptiert Dr. Logo eine Zahl mit einem Anführungszeichen überall dort, wo er normalerweise eine Zahl ohne Anführungszeichen erwartet. Er akzeptiert auch Zahlen ohne Anführungszeichen überall dort, wo er normalerweise ein Wort mit Anführungszeichen erwartet. Ein Kernbefehl, der Zahlen als Eingabeparameter erwartet, akzeptiert zum Beispiel auch Zahlen mit Anführungszeichen:

```
?fd "50
?+ "53 "42
95
?
```

Ein Kernbefehl, der ein Wort oder eine Liste erwartet, akzeptiert auch Zahlen ohne Anführungszeichen:

```
?first 9876
9
?
```

Eine Ausnahme von dieser Regel stellt der Kernbefehl `ascii` dar, er erwartet ein Zeichen mit vorangestelltem Anführungszeichen als Eingabeparameter. `ascii` macht zwischen Ziffern, Buchstaben und Symbolen keinen Unterschied.

```
?ascii "2
50
?ascii 2
ascii doesn't like 2 as input
?
```

Sie können einer Zahl ein + oder ein - voranstellen, um ihr Vorzeichen festzulegen. Da aber diese Zeichen sowohl Begrenzer und arithmetische Operatoren als auch Vorzeichen darstellen, können Sie zufällig und ungewollt aus zwei Zahlen einen Ausdruck zusammenstellen. Um dies zu vermeiden, interpretiert Dr. Logo Leerzeichen und Zahlen zusammen mit arithmetischen Operatoren auf folgende Art und Weise:

```
5-2    wird als 3 interpretiert.
5 - 2  wird auch als 3 interpretiert.
5+-2   wird auch als 3 interpretiert, allerdings wird
5 -2   als zwei Zahlen, 5 und -2, interpretiert.
```

Sie können negative oder positive Dezimalzahlen mit bis zu 15 signifikanten Ziffern eingeben. Dr. Logo gibt die Zahl in technisch-wissenschaftlicher Darstellung zurück, er stellt den Kleinbuchstaben `e` zwischen die Dezimalzahl und die Zehnerpotenz. So zum Beispiel:

```
?1000000000000000000
1.e+016
?
```

Die Zehnerpotenz können Sie sich als Anzahl der Dezimalstellen vorstellen, um die der Dezimalpunkt zu bewegen ist. Wenn die Zahl hinter dem `e` positiv ist, wird der Dezimalpunkt nach rechts verschoben. Ist die Zahl hinter dem `e` negativ, wird der Dezimalpunkt nach links verschoben.

```
? .000000000000000001
1.e-016
?
```

Große Dezimalzahlen können Sie Dr. Logo in technisch-wissenschaftlicher Darstellung eingeben. Verwenden Sie ein großes `E` oder ein kleines `e`, um die Zehnerpotenz zu identifizieren.

```
?1.23e15 + 3.21E10
1.2300321e+015
?
```

Die meisten arithmetischen Operationen akzeptieren sowohl Dezimal- als auch ganze Zahlen als Eingabeparameter. Einige Operationen akzeptieren zwar auch ganze Zahlen als Eingabeparameter, aber geben eine Dezimalzahl aus.

```
?13/7
1.85714285714286
?
```

Dr. Logo unterstützt Kernbefehle, die Dezimalzahlen in ganze Zahlen umformen, Zufallszahlen erzeugen und trigonometrische Funktionswerte berechnen. Intern verwendet Dr. Logo ganze Zahlen und Fließpunktzahlen sowohl in einfacher als auch doppelter Genauigkeit. In seinen Berechnungen verwendet Dr. Logo ausschließlich Zahlen doppelter Genauigkeit, aber er formt diese Zahlen zur Ausgabe oder Abspeicherung zu Zahlen einfacher Genauigkeit um, wenn die Genauigkeit von 15 signifikanten Stellen dadurch nicht vermindert wird.

## 2.2.4 Variablen\*

Eine Variable ist ein *Behälter*, der ein Argument enthält. Ein Behälter in Ihrer Küche kann Schokolade, Erdnußbutter, andere Süßigkeiten oder gar keine Süßigkeiten enthalten. Eine Variable kann einen beliebigen Namen besitzen, den Sie ihr gegeben haben, und sie kann beliebige Argumente enthalten. Ein Argument ist ein Wort, eine Zahl oder eine Liste. Hier ist die einfachste Methode, um eine Variable zu erzeugen:

```
?make "Lieblinge [Schokolade Erdnussbut!
ter Plätzchen Kekse Gummibärchen Luts!
hbonbon]
?
```

Nachdem der Kernbefehl `make` ausgeführt wurde, hat die lange Liste der Süßigkeiten einen Namen: "Lieblinge. Beachten Sie das Ausrufungszeichen im obigen Beispiel. Wenn Sie eine Zeile eingeben, die die Breite Ihres Monitors überschreitet, gibt Dr. Logo ein Ausrufungszeichen aus und setzt die Zeile in der nächsten Bildschirmzeile fort. Damit können Sie einen langen Ausdruck eingeben, ohne die Return-Taste zu drücken, bis Sie mit dem langen Ausdruck fertig sind, den Dr. Logo auswerten soll.

Die Möglichkeit, auf ein Objekt durch den Variablennamen Bezug zu nehmen, ermöglicht Ihnen, einen Ausdruck zu schreiben, der unabhängig von den Objekten ist, die er manipuliert. So kann die für das Süßigkeitenfach verantwortliche Person zum Beispiel folgendes eingeben:

```
?make "Süßigkeit.fach []
?if empty :Süßigkeit.fach [make "Süssi!
keit.fach first shuffle :Lieblinge]
?
```

D.h., wenn das Süßigkeitenfach leer ist, fülle die erste Süßigkeitenart aus der gemischten Liste der Lieblinge in das Süßigkeitenfach.

Es gibt mehrere Ausdrücke, die das Verhältnis zwischen Namen und Inhalt einer Variablen beschreiben. Wenn der Ausdruck `make` zum Beispiel `Lutschbonbon` in `Süssigkeit.fach` füllt, kann man das Verhältnis zwischen `Lutschbonbon` und `Süssigkeit.fach` mit folgenden Aussagen beschreiben:

- o `Süssigkeit.fach` ist eine Variable, `Lutschbonbon` ist ihr Wert.
- o `Süssigkeit.fach` ist der Name von `Lutschbonbon`.
- o `Süssigkeit.fach` enthält `Lutschbonbon`.
- o `Süssigkeit.fach` ist mit `Lutschbonbon` verbunden.
- o `Lutschbonbon` ist das *Ding* von `Süssigkeit.fach`.
- o `Lutschbonbon` ist der Inhalt von `Süssigkeit.fach`.
- o `Lutschbonbon` ist der Wert von `Süssigkeit.fach`.

Es gibt zwei verschiedene Methoden, um auf den Inhalt einer Variablen Bezug zu nehmen:

```
?thing "Süssigkeit.fach
Lutschbonbon
?:Süssigkeit.fach
Lutschbonbon
?
```

Der Doppelpunkt vor dem Variablennamen sagt Dr. Logo, daß er den Inhalt der Variablen statt ihren Namen als ein Wort betrachten soll. Lesen Sie obiges Beispiel als *den Wert von Süssigkeit.fach* oder *der Inhalt von Süssigkeit.fach*.

Unabhängig davon, was Sie über Variablen denken, prägen Sie sich ein, daß Sie Variablen in Ausdrücken verwenden können, ohne sich um den aktuellen Wert der Variablen kümmern zu müssen. Die Variablen helfen Ihnen, Ausdrücke zu schreiben, die unabhängig von den Daten sind, die Sie bearbeiten. Kapitel 4 *Arbeiten mit Prozeduren* zeigt Ihnen, wie Sie Variablen in Prozeduren verwenden können.

### 2.3 Zeilen und Ausdrücke

Sie können so viele Ausdrücke in einer Zeile angeben, wie Sie wollen, aber eine Zeile in der Hauptebene (d.h. nach dem Prompt `?`) kann nicht mehr als 133 Zeichen enthalten (inklusive Prompt und Ausrufezeichen).

Normalerweise verfährt Dr. Logo bei der Auswertung der Ausdrücke in einer Zeile nach dem Von-links-nach-rechts-Schema. Dr. Logo betrachtet alles, was rechts vom Prozedurnamen oder Kernbefehl steht (einschließlich anderer Ausdrücke), als Eingabeparameter zur betreffenden Prozedur, sofern diese irgendwelche Eingaben erwartet.

```
?fd 50 rt (80 + 10) bk 50
?
```

Arithmetische Ausdrücke wertet Dr. Logo nicht streng von links nach rechts aus. Er wertet zuerst die Operationen `/` und `*` von links nach rechts aus, dann geht er zurück und wertet die Operationen `+` und `-` aus:

```
?? * 3 + 7 / 5
7.4
?
```

Um Dr. Logo einen Ausdruck in einer anderen Reihenfolge auswerten zu lassen, können Sie Ausdrücke in Klammern gruppieren. Dr. Logo wertet zuerst den Ausdruck in den innersten Klammern aus. Die Reihenfolge, die Dr. Logo bei der Auswertung eines Ausdruckes verwendet, kann zu großen Unterschieden führen!

```
?? * 3 + 7 / 5
7.4
?? * (3 + 7) / 5
4
?
```

## 3 Editier-Kommandos

Mit Dr. Logo zu editieren bedeutet einen neuen Text einzugeben oder einen früher eingegebenen Text zu verändern. Editieren kann sowohl ganz einfach eine Korrektur eines Tippfehlers sein als auch der komplizierte Vorgang des Definierens langer Prozeduren in Dr. Logos Textpuffer-Editor.

In Dr. Logo gibt es drei verschiedene Editier-Möglichkeiten: *Zeileneditieren*, *Prozedureditieren* und *Texteditieren*. Die Kommandos des Zeileneditierens können Sie auch bei Prozedureditieren und Texteditieren verwenden. Prozedureditieren ist eine einfache Erweiterung des Zeileneditierens; das Texteditieren ist auf dem Prozedureditieren aufgebaut. Dieses Kapitel zeigt Ihnen, wann, warum und wie Sie Dr. Logos Zeilen-, Prozedur- und Texteditier-Kommandos verwenden können.

Die in diesem Kapitel beschriebenen Kommandos geben Sie Dr. Logo nicht mit Hilfe von Ausdrücken, sondern mit Kontrollzeichen oder Funktionstasten ein. Nicht alle Kontrollzeichenkommandos von Dr. Logo dienen zum Texteditieren, sondern einige unterbrechen oder beenden die Ausführung einer Prozedur. Der letzte Teil in diesem Kapitel enthält eine Einführung zu den verschiedenen Kontrollzeichenkommandos von Dr. Logo.

Um ein Kontrollzeichen einzugeben, halten Sie die Kontrolltaste (mit *Control* bzw. *Alt* markiert) gedrückt und drücken die erwünschte Buchstabentaste. Dr. Logo verfügt über Sondertasten und Funktionstasten, die die meisten Kontrollzeichen ersetzen können.

### 3.1 Zeileneditieren

Die Zeileneditier-Kommandos können Sie zum Korrigieren beliebiger Texte verwenden, die Sie Dr. Logo eingeben. Sie können diese Kommandos benutzen, wenn Sie in der Hauptebene arbeiten (d.h. wenn Sie Antworten auf das Prompt ? eingeben) oder wenn Sie Ihre Programme mit Hilfe des Kommandos *watch* austesten. Sie können diese Kommandos sogar während der Ausführung einer Prozedur verwenden. So zum Beispiel, wenn ein Programm Sie während der Ausführung zur Eingabe irgendwelcher Daten über die Tastatur auffordert, können Sie Ihre Antwort durch Anwendung der Zeileneditier-Kommandos verändern. In den Tabellen 3-1a und 3-1b sind Dr. Logos Zeileneditier-Kommandos für den CPC6128 und den Joyce zusammengefaßt. Alternative Eingaben sind in Klammern angegeben.

Tabelle 3-1a: Kommandos zum Zeileneditieren beim CPC6128

Kommando	Operation
→ (Control-F)	Der Cursor springt um ein Zeichen vorwärts, d.h. um eine Position
← (Control-B)	Der Cursor springt um ein Zeichen zurück; d.h. um eine Position nach links.
Control→ (Control-E)	Der Cursor springt zum Ende der Zeile.
Control← (Control-A)	Der Cursor springt zum Anfang der Zeile.
Clr (Control-D)	Löscht das Zeichen, das durch den Cursor markiert ist.
Del (Control-H)	Löscht das Zeichen links vom Cursor.
Tab (Control-I)	Fügt einen Tabulator ein. Der Cursor springt zum nächsten Tabulator, der in der betreffenden Zeile gesetzt wurde.
Control-K	Löscht den Rest der Zeile, d.h. es löscht alle Zeichen rechts vom Cursor bis zu 133 Zeichen. Die gelöschten Zeichen werden im Zeileneditier-Puffer gespeichert und können mit Control-Y wiederhergestellt werden.
Control-Y	Stellt einen Text aus dem Puffer wieder her, d.h. die letzte in den Puffer mit der Return-Taste oder mit Control-K eingespeicherte Zeile wird wieder angezeigt.

Tabelle 3-1b: Kommandos zum Zeileneditieren beim Joyce

Kommando	Operation
→ (Alt-F)	Der Cursor springt um ein Zeichen vorwärts, d.h. um eine Position
← (Alt-B)	Der Cursor springt um ein Zeichen zurück; d.h. um eine Position nach links.
[Zeile Ez] (Alt-E)	Der Cursor springt zum Ende der Zeile.
Shift-[Zeile Ez] (Alt-A)	Der Cursor springt zum Anfang der Zeile.
[Del→] (Alt-D)	Löscht das vom Cursor markiert Zeichen
[+Del] (Alt-H)	Löscht das Zeichen links vom Cursor
Tab (Alt-I)	Fügt einen Tabulator ein. Der Cursor springt zum nächsten Tabulator, der in der betreffenden Zeile gesetzt wurde.
Alt-[Del→] (Alt-K)	Löscht den Rest der Zeile, d.h. es löscht bis zu 133 Zeichen rechts vom Cursor. Die gelöschten Zeichen werden im Zeileneditier-Puffer gespeichert und können mit Alt-Y wiederhergestellt werden.
Copy, Einbl (Alt-Y)	Stellt einen Text aus dem Puffer wieder her, d.h. die letzte in den Puffer mit der Return-Taste oder mit Alt-K eingespeicherte Zeile wird wieder angezeigt.

Das nächste Beispiel zeigt Ihnen, wie Sie die Funktionstasten und diese Kontrollzeichen verwenden können, wenn Sie Ausdrücke für den Interpretierer eingeben.

Nachdem Sie eine Zeile eingegeben haben, können Sie verschiedene Kontrollzeichen verwenden, um den Cursor im Text links und rechts zu bewegen. Sie können damit überall im Text Korrekturen machen, bevor Sie die Return-Taste drücken.

Wirkung	Tätigkeit
?repaet 36 [fd 8 Lt 10]	repeat und lt falsch eingegeben
?repaet 36 [fd 8 Lt 10]	Control-→ bzw. Shift-[Zeile Ez] zum Zeilenanfang
?repaet 36 [fd 8 Lt 10]	3 mal →, um den Cursor um 3 Positionen nach rechts zu bewegen
?repaet 36 [fd 8 Lt 10]	Clr bzw. Del+, löscht Zeichen unter dem Cursor
?repaet 36 [fd 8 Lt 10]	1 mal →, um den Cursor um eine Position nach rechts zu bewegen
?repaet 36 [fd 8 Lt 10]	a eingegeben, um repeat zu korrigieren
?repeat 36 [fd 8 Lt 10]	Control-→ bzw. [Zeile Ez], um den Cursor zum Zeilenende zu bewegen
?repeat 36 [fd 8 Lt 10]	5 mal ←, um den Cursor um 5 Position nach links zu bewegen
?repeat 36 [fd 8 Lt 10]	Del bzw. [+Del], um das Zeichen links von der Cursorposition zu löschen
?repeat 36 [fd 8 Lt 10]	1 eingegeben, um lt zu korrigieren

Unabhängig von der aktuellen Position des Cursors können Sie jederzeit die Return-Taste verwenden, um Ihr Kommando an Dr. Logo abzuschicken. Wenn Sie die Return-Taste drücken, um eine Zeile dem Interpretierer einzugeben, speichert Dr. Logo diese Zeile im Zeilenpuffer ab. Dr. Logo speichert im gleichen Zeilenpuffer auch die letzte Zeile, die mit Control-K bzw. Alt-[Del+] gelöscht wurde. Diese zuletzt gelöschte Zeile können Sie mit Control-Y bzw. Copy oder Einbl wiederherstellen. Diese Möglichkeit ist dann sehr nützlich, wenn Sie ein Kommando mehrmals ausführen lassen wollen, wenn Sie ein Kommando mit kleinen Änderungen wieder ausführen lassen wollen, oder wenn Sie einen Tippfehler gemacht haben und die Return-Taste gedrückt haben, bevor Sie den Fehler bemerkt und korrigiert haben.

Wirkung	Tätigkeit
?save bilder I don't know how to bilder ?	
?save bilder	Control-Y bzw. Copy oder Einbl stellt die Zeilewieder her
?save "b"bilder ?	6 mal →, um den Cursor nach links zu bewegen; das eingefügte Anführungszeichen korrigiert die Zeile
? save "bilder"	Return drücken, um die Datei auf dem Standardlaufwerk abzuspeichern
? save "b"bilder	Control-Y bzw. Copy oder Einbl stellt die Zeile wieder her
?save "b"bilder	6 mal ←, um den Cursor nach links zu bewegen
save "b:b"bilder ?	b: eingegeben und Return gedrückt, um eine Kopie auf der Diskette im Laufwerk B bzw. auf Seite B zu erstellen
?	
Mit Control-K bzw. Alt-[Del+] können Sie eine ganze Zeile oder einen Teil der Zeile löschen. Wenn Sie es sich später anders überlegen, können Sie die gelöschten Zeichen mit Control-Y bzw. Copy oder Einbl wiederherstellen.	
Wirkung	Tätigkeit
?ern "bilder.paket"	Namen löschen? Vielleicht doch nicht...
?ern "bilder.paket"	mit Control-← bzw. Shift-[Zeile Ez] zum Zeilenanfang
? ?	Control-K bzw. Alt-[Del+] löscht die Zeile
?ern "bilder.paket"	Ja, Namen löschen; Control-Y bzw. Copy oder Einbl stellt die Zeile wieder her
? ?	

Sie können Control-K bzw. Alt-[Del→] und Control-Y bzw. Copy oder Einbl auch zur Wiederholung eines Teiles der Zeile verwenden.

Wirkung	Tätigkeit
?(pr "I char 43 "U " I + U	
?(PR "I char 43 "U "■	Control-Y bzw. Copy oder Einbl stellt die Zeile wieder her
?(pr "I char 43 "U "■	Control-← bzw. Shift-[Zeile Ez] und 4 mal →, um den Cursor nach links zu bewegen
?(pr ■	Control-K bzw. Alt-[Del→] löscht einen Teil der Zeile
?(pr "I char 43 "U "■	Control-Y bzw. Copy oder Einbl stellt den vorher gelöschten Teil der Zeile wieder her
?(pr "I char 43 "U " "I char 43 "U " "I char 3 "U "■	Control-Y bzw. Copy, Leerzeichen, Control-Y bzw. Copy, Leerzeichen erweitern die Kommandozeile
I + U    I + U    I + U ?	

### 3.2 Prozedureditieren

Wenn Sie das Prompt ? beantworten, oder wenn Sie sich während der Ausführung einer Prozedur in einer pause befinden, können Sie die spezifischen Worte to oder end verwenden, um den Prozedureditor aufzurufen oder ihn zu verlassen. Mit anderen Worten, wenn Sie als Antwort auf das Prompt ? oder während einer pause eine Zeile eingeben, die mit to beginnt, ruft Dr. Logo den Prozedureditor auf.

```
?to kreis :grösse
>repeat 36 [fd :grösse rt 10]
>end
kreis definiert
?
```

Das Prompt des Prozedureditors, >, zeigt Ihnen, daß Dr. Logo die von Ihnen zur Definition der neuen Prozedur eingegebenen Ausdrücke nicht sofort auswerten wird.

Wenn Sie den Prozedureditor aufgerufen haben, können Sie, um den Cursor zu bewegen, Fehler auszubessern, Zeichen und Zeilen zu löschen und wiederherzustellen, die Zeileneditier-Kommandos verwenden.

Um den Prozedureditor zu verlassen, müssen Sie das Schlüsselwort end am Anfang der letzten Zeile der Prozedur allein eingeben. Nachdem er die end-Zeile gelesen hat, definiert Dr. Logo die Prozedur und kehrt in den Zustand zurück, von woher Sie den Prozedureditor aufgerufen haben, also entweder zur Hauptebene (Prompt ?) oder zur pause während der Ausführung einer Prozedur.

### 3.3 Texteditieren

Um in einer bereits früher definierten Prozedur Änderungen vorzunehmen, ohne zum ganzen Vorgang des Definierens in den Prozedureditor zurückzukehren, können Sie Dr. Logos Texteditor benutzen. Wenn Sie Dr. Logos Texteditor benutzen, können Sie sich von Zeile zu Zeile in der Prozedur fortbewegen, und in jeder Zeile Änderungen vornehmen. Sie können eine beliebige Anzahl von Prozeduren in den Texteditor laden und in allen diesen Prozeduren Änderungen vornehmen. Außerdem können Sie den Texteditor auch zum Verändern des Inhalts von Variablen benutzen.

Der Texteditor macht bestimmte Annahmen bezüglich Ihrer Ziele für eine Editiersitzung. Wenn Sie zum Beispiel eine Prozedur im Texteditor definieren und danach vergessen die end-Zeile am Ende einzugeben, fügt der Texteditor eine end-Zeile zu Ihrer Prozedur hinzu. Wenn eine Fehlermeldung während der Ausführung einer Ihrer Prozeduren ausgegeben wird und Sie den Texteditor aufrufen, nimmt Dr. Logo an, daß Sie diese Prozedur editieren wollen, er lädt automatisch diese Prozedur in den Editierpuffer und stellt den Cursor auf die Zeile, in der der Fehler aufgetreten ist.

Ähnlich, wie beim Prozedureditor, können Sie sowohl in der Hauptebene (Prompt ?), als auch während einer pause oder während der Beobachtung der Ausführung einer Prozedur mit watch den Texteditor aufrufen.

Bevor Sie den Texteditor zur Änderungen in Ihren Prozeduren oder Variablen benutzen können, müssen Sie die zu ändernden Objekte in den Editierpuffer laden. Der Editierpuffer ist ein eigener Bereich des Texteditors im Hauptspeicher. Innerhalb dieses Puffers können Sie Änderungen vornehmen, Texte hinzufügen und Texte löschen. Diese Änderungen werden allerdings nicht in den Arbeitsspeicher von Dr. Logo übertragen, solange Sie nicht zum Abspeichern des Pufferinhalts Control-C drücken. Bevor Sie die Änderungen auf einer Diskette abspeichern können, müssen diese Änderungen unbedingt in den Arbeitsspeicher von Dr. Logo übertragen werden. Wenn Sie den Arbeitsspeicher nicht ändern möchten, drücken Sie Control-G.

Es gibt zwei Kernbefehle, die Prozeduren und Variablen in den Editierpuffer laden: ed und edall. Der einzige Unterschied zwischen diesen Kernbefehlen besteht darin, was sie in den Editierpuffer laden und, wenn überhaupt, welche Eingabenamen sie

verlangen, um ausgewählte Prozedur- und/oder Variablengruppen in den Editierpuffer zu laden.

**ed** können Sie zum Laden der angegebenen Variable(n) oder Prozedur(en) in den Texteditorpuffer verwenden. Als Eingabe akzeptiert ed Prozeduren- bzw. Variablennamen mit Anführungszeichen, oder eine Liste von Prozeduren- bzw. Variablennamen. Wenn die Ausführung einer Prozedur mit der Ausgabe einer Fehlermeldung beendet wurde und Sie unmittelbar danach ed ohne Prozedurnamen aufrufen, lädt der Texteditor automatisch die fehlerhafte Prozedur in den Texteditorpuffer und setzt den Cursor in die fehlerhafte Zeile.

**edall** können Sie zum Laden aller Variablen und Prozeduren aus dem Arbeitsspeicherbereich in den Texteditorpuffer verwenden.

Wenn Sie einen dieser Kernbefehle zum Aufrufen des Texteditors benutzen, löscht Dr. Logo allen Text vom Bildschirm und in der untersten Zeile des leeren Bildschirms erscheint eine Meldung:

EDIT

Der Texteditor gibt kein Promptzeichen aus, aber der Cursor zeigt die Stelle an, die durch ein Kontrollzeichenkommando oder ein Einfügeprozess berührt werden wird.

Solange Sie sich im Texteditor befinden, können Sie zwecks Korrektur innerhalb der einzelnen Zeilen alle Funktionstasten und Kontrollzeichen des Zeileneditierens anwenden, die im Abschnitt 3.1 beschrieben wurden. In Tabelle 3-2 sind die Kontrollzeichen beim Texteditieren zusammengefaßt, die Sie zum Bewegen des Cursors zwischen Zeilen, bzw. Seiten benutzen und mit denen Sie den Texteditor verlassen können.

**Tabelle 3-2a: Kommandos zum Texteditieren beim CPC6128**

Kommando	Definition
<b>Copy</b> (Control-C)	Beendet das Texteditieren, verläßt den Texteditor und aktualisiert den Arbeitsspeicherbereich von Dr. Logo mit allen Prozeduren und Variablen aus dem Texteditorpuffer. Wenn Kommandos im Editierpuffer abgespeichert sind, werden sie ausgeführt.

**Esc**  
(Control-G)

Verläßt den Texteditor, läßt aber den Arbeitsspeicherbereich von Dr. Logo unverändert. Ignoriert den ganzen Editierpuffer, einschließlich aller Änderungen, die Sie während der Editiersitzung vorgenommen haben. Sie können den verlassenen Editierpuffer durch Eingabe von ed wiederherstellen, wenn Sie nach Drücken der Esc-Taste noch kein anderes Zeichen eingegeben haben.

**Control-L**

Stellt die Bildschirmanzeige so ein, daß die Zeile, in der sich der Cursor befindet, in der Mitte des Bildschirms erscheint. Wenn die Entfernung zwischen dem Pufferanfang und dem Cursor kleiner ist, als 12 Zeilen, wird das Textfenster beim Drücken von Control-L einfach erneut ausgegeben.

**↓**  
(Control-N)

Der Cursor springt zur nächsten Zeile; er bewegt sich um eine Zeile nach unten in Richtung zum Ende des Puffers.

**Control-O**

Schafft eine neue Zeile. Eingabe von Control-O entspricht der unmittelbar aufeinanderfolgenden Eingabe der Return-Taste und ↑.

**↑**  
(Control-P)

Der Cursor springt um eine Zeile zurück, d.h. nach oben, in der Richtung zum Anfang des Puffers

**Control-↓**  
(Control-V)

Gibt die nächste Seite des Textes im Texteditorpuffer auf dem Bildschirm aus, d.h. die nächsten Zeilen in Richtung zum Ende des Puffers.

**Control-↑**  
(Control-U)

Gibt die vorherige Seite des Textes im Texteditorpuffer auf dem Bildschirm aus, d.h. die vorherigen Zeilen in Richtung zum Anfang des Puffers.

**Control-R**

Der Cursor springt zum Anfang des Texteditorpuffers.

**Control-X** Der Cursor springt zum Ende des Texteditorpuffers.

Tabelle 3-2b: Kommandos zum Texteditieren beim Joyce

Kommando	Definition
<b>Exit</b> (Alt-G)	Beendet das Texteditieren, verläßt den Texteditor und aktualisiert den Arbeitsspeicherbereich von Dr. Logo mit allen Prozeduren und Variablen aus dem Texteditorpuffer. Wenn Kommandos im Editierpuffer abgespeichert sind, werden sie ausgeführt.
<b>Stop</b> (Alt-G)	Verläßt den Texteditor, läßt aber den Arbeitsspeicherbereich von Dr. Logo unverändert. Ignoriert den ganzen Editierpuffer, einschließlich aller Änderungen, die Sie während der Editiersitzung vorgenommen haben. Sie können den verlassenen Editierpuffer mit <code>ed</code> wiederherstellen, wenn Sie nach der Stop-Taste noch kein anderes Zeichen eingegeben haben.
<b>Alt-L</b>	Stellt die Bildschirmanzeige so ein, daß die Zeile, in der sich der Cursor befindet, in der Mitte des Bildschirms erscheint. Wenn die Entfernung zwischen dem Pufferanfang und dem Cursor kleiner ist, als 12 Zeilen, wird das Textfenster beim Drücken von Alt-L einfach erneut ausgegeben.
<b>↓</b> (Alt-N)	Der Cursor springt zur nächsten Zeile; er bewegt sich um eine Zeile nach unten in Richtung zum Ende des Puffers.
<b>Alt-O</b>	Schafft eine neue Zeile. Eingabe von Alt-O entspricht der unmittelbar aufeinanderfolgenden Eingabe von Return und ↑.
<b>↑</b> (Alt-P)	Der Cursor springt um eine Zeile zurück, d.h. nach oben, in der Richtung zum Anfang des Puffers

[Seite]  
(Alt-V) Gibt die nächste Seite des Textes im Texteditorpuffer auf dem Bildschirm aus, d.h. die nächsten Zeilen in Richtung zum Ende des Puffers.

Alt-[Seite]  
(Alt-U) Gibt die vorherige Seite des Textes im Texteditorpuffer auf dem Bildschirm aus, d.h. die vorherigen Zeilen in Richtung zum Anfang des Puffers.

Alt-Shift-[Dok]  
(Alt-R) Der Cursor springt zum Anfang des Texteditorpuffers.

Shift-[Dok]  
(Alt-X) Der Cursor springt zum Ende des Texteditorpuffers.

Das nächste Beispiel zeigt, wie Sie diese Kontrollzeichen beim Texteditieren mit denen beim Zeileneditieren kombinieren können, um die Editiersitzung zu beschleunigen. Einige Kontrollzeichen-Kombinationen und Funktionstasten sind sehr nützlich, Sie sollten diese Kombination gut einprägen. Die Kombination `Control-→` und `Clr` bzw. `[Ez]` und `[Del→]` löscht den Wagenrücklauf zwischen zwei Zeilen und macht aus zwei Zeilen eine Zeile. `Control-+` und `Control-K` bzw. `Shift-[Ez]` und `[Del→]` löscht die Zeile, in der sich der Cursor befindet. Sie können `Control-O` mit `Control-Y` oder `Control-K` bzw. `Copy` oder `[Del→]` kombinieren, um eine Zeile innerhalb des Editors schnell zu verschieben.

Die Zeile, die im Beispiel unten verschoben wird, läßt die Prozedur abzählen vor Beendigung ihrer Arbeit eine Null ausgeben. Die ganze Editiersitzung wird mit nur sieben Kommandos durchgeführt.

Wirkung Tätigkeit

```
?to abzählen :nummer
>if :nummer = 0 [stop]
>pr :nummer
>abzählen :nummer-1
>end
abzählen defined
?abzählen 4
4
3
2
1
?ed "abzählen
```

```
to abzählen :nummer
if :nummer = 0 [stop]
pr :nummer
abzählen :nummer-1
end
```

Der Editor löscht den Bildschirminhalt

```
to abzählen :nummer
if :nummer = 0 [stop]
pr :nummer
abzählen :nummer-1
end
```

↓ und Control+ bzw. Shift-[Ez] bewegen den Cursor in die Zeile, die verschoben werden soll

```
to abzählen :nummer
pr :nummer
abzählen :nummer-1
end
```

Control-K bzw. Alt-[Del+] löscht die Zeile

```
to abzählen :nummer
pr :nummer
abzählen :nummer-1
end
```

2 mal ↓, um den Cursor auf die neue Stelle zu positionieren

```
to abzählen :nummer
pr :nummer
if :nummer = 0 [stop]
abzählen :nummer-1
end
```

Control-O, Control-Y bzw. Copy oder Einbl fügt eine Zeile ein

```
abzählen defined
?abzählen 4
4
3
2
1
0
?
```

Copy bzw. Exit definiert die Prozedur

Die Funktionstasten und Kontrollzeichenkommandos zum Seitenwechsel sind dann nützlich, wenn Sie mehr Text im Texteditorpuffer haben, als Sie auf einmal auf dem Bildschirm ansehen können. Sie können mit `eda11` alles aus Dr. Logos Arbeitsspeicherbereich in den Texteditorpuffer laden und dann experimentieren.

Um den Editor auf die übliche Weise zu verlassen, drücken Sie beim CPC6128 Copy und Exit beim Joyce. In diesem Fall wird Dr. Logo die Definition der Prozeduren und Variablen in seinem Arbeitsspeicherbereich durch die entsprechenden Definitionen im Texteditorpuffer ersetzen. Wenn Sie die Titelzeile einer Prozedur geändert haben, definiert Dr. Logo eine neue Prozedur und läßt die ursprüngliche Definition der Prozedur, die sich im Arbeitsspeicherbereich befindet, ungeändert. Wenn Sie am Ende einer Prozedur die `end`-Zeile vergessen haben, ergänzt Dr. Logo die Prozedur mit dieser Zeile, während er die ursprüngliche Prozedur in seinem Arbeitsspeicherbereich durch die Prozedur im Texteditorpuffer ersetzt.

Wenn Sie die während der Editiersitzung vorgenommenen Änderungen ignorieren möchten, verlassen Sie den Texteditor mit beim CPC6128 mit Esc und mit Stop beim Joyce. Wenn Sie es sich dann doch anders überlegen, können Sie wieder `ed` eingeben, um den verlassenen Editierpuffer wiederherzustellen.

Wenn Sie den Texteditor verlassen, kehren Sie dorthin zurück, von woher Sie den Editor aufgerufen haben. Wenn Sie den Texteditor zum Beispiel während einer pause in einer Prozedur angerufen haben, gibt Dr. Logo nach Verlassen des Texteditors das entsprechende `pause`-Prompt aus und wartet auf Ihr nächstes Kommando. Sie können einige Texteditierkommandos auch außerhalb des Texteditors verwenden. Wenn Sie einen langen Ausdruck außerhalb des Editors eingeben und Dr. Logo ein Ausrufezeichen ausgibt, bevor er in die nächste Zeile springt, können Sie ↑ und ↓ benutzen, um den Cursor zwischen den Zeilen auf und ab zu bewegen.

### 3.4 Weitere Kontrollzeichenkommandos und Funktionstasten

Dr. Logo kennt auch einige Kontrollzeichenkommandos und Funktionstasten für Tätigkeiten außer Texteditieren. Manche Kontrollzeichenkommandos haben verschiedene Bedeutungen abhängig davon, ob Sie sich im Editor oder in der Hauptebene befinden. In diesem Abschnitt werden die Kontrollzeichen diskutiert, die zum Abbrechen oder Unterbrechen der Ausführung von Prozeduren, zum Rollen des Textes auf dem Bildschirm und zum Umschalten zwischen Vollgraphik-, Volltext- und geteilter Bildschirmanzeige dienen. Die Bildschirm-Kontrollzeichenkommandos sind in Kapitel 5 ausführlich beschrieben. In Tabelle 3-3 sind die verschiedenartigen Kontrollzeichenkommandos zusammengefaßt.

Tabelle 3-3a: Zusätzliche Kommandos beim CPC6128

Kommando	Operation
Esc (Control-G)	Sofortige Beendigung der Ausführung der aktuellen Prozedur.
Control-L	Ausgabe einer vollen Zeichenfläche; der Bildschirm steht für die Igelgraphik allein zur Verfügung.
Control-M	Erzeugt einen Wagenrücklauf (Return); gibt die Information in den Rechner ein.
Control-Q	Generiert ebenso wie die entsprechende Taste den umgekehrten Schrägstrich (\), der Dr. Logo einen Begrenzer als Textbestandteil betrachten läßt.
Control-S	Teilt den Bildschirm; eröffnet ein Textfenster auf dem Graphikbildschirm.
Control-T	Schaltet in Textmodus um: der Bildschirm dient zur Textausgabe. Drücken Sie zweimal Control-T, damit der Cursor auch oben bleibt!
Control-W	Unterbricht das Rollen eines Textes; wartet bis zum Drücken der nächsten Taste, um die Ausgabe fortzusetzen.
Control-Z	Unterbricht die momentan ausgeführte Prozedur und gibt das pause-Prompt aus, um interaktive Fehlersuche zu ermöglichen. Geben Sie <code>co</code> ein, wenn Sie die Ausführung der Prozedur fortsetzen möchten; geben Sie <code>throw "TOPLEVEL</code> ein, wenn Sie zur höchsten Ebene umsteigen möchten oder geben Sie <code>stop</code> ein, wenn Sie zur vorherigen Ebene zurückkehren möchten.

Tabelle 3-3b: Zusätzliche Kommandos beim Joyce

Kommando	Operation
Stop (Alt-G)	Sofortige Beendigung der Ausführung der aktuellen Prozedur.
Alt-L	Ausgabe einer vollen Zeichenfläche; der Bildschirm steht für die Igelgraphik allein zur Verfügung.
Alt-M	Erzeugt einen Wagenrücklauf (Return); gibt die Information in den Rechner ein.
Alt-Q	Generiert ebenso wie die entsprechende Taste den Großbuchstaben Q, der Dr. Logo einen Begrenzer als Textbestandteil betrachten läßt.
Alt-S	Teilt den Bildschirm; eröffnet ein Textfenster auf dem Graphikbildschirm.
Alt-T	Schaltet in Textmodus um: der Bildschirm dient zur Textausgabe. Drücken Sie zweimal Alt-T, damit der Cursor nach oben geht!
Alt-W	Unterbricht das Rollen eines Textes; wartet bis zum Drücken der nächsten Taste, um die Ausgabe fortzusetzen.
Alt-Z	Unterbricht die momentan ausgeführte Prozedur und gibt das pause-Prompt aus, um interaktive Fehlersuche zu ermöglichen. Geben Sie <code>co</code> ein, wenn Sie die Ausführung der Prozedur fortsetzen möchten; geben Sie <code>throw "TOPLEVEL</code> ein, wenn Sie zur höchsten Ebene umsteigen möchten oder geben Sie <code>stop</code> ein, wenn Sie zur vorherigen Ebene zurückkehren möchten.

## 4 Arbeiten mit Prozeduren

Dieses Kapitel diskutiert, wie man eine Prozedur zusammenstellt, wie man mehrfache Eingabeparameter eingibt und wie Dr. Logo die Ausführung der Prozeduren überwacht.

### 4.1 Benennung und Definition der Prozeduren

Eine Prozedur zu definieren bedeutet Dr. Logo eine neue Operation beizubringen. Die neue Prozedur können Sie mit Kernbefehlen oder anderen Wörtern aus Dr. Logos Vokabular beschreiben. Dr. Logo kennt zum Beispiel Kernbefehle; die den Igel vorwärts (fd) bewegen, eine Bewegung wiederholen (repeat) oder eine Rechtsdrehung durchführen (rt) lassen. Durch Anwendung dieser Kernbefehle können Sie Dr. Logo beibringen, wie er ein Dreieck zeichnen kann. Danach können Sie die Prozedur dreieck zur Definition einer anderen Prozedur, wie zum Beispiel flage verwenden.

Die Definition einer neuen Prozedur fängt mit Eingabe des Schlüsselwortes to als Antwort auf das Prompt ? an. Dieses Wort sagt Dr. Logo, daß er das nächste Wort, das Sie eingeben, als den Namen der neuen Prozedur interpretieren soll. Prozedurnamen können mit einem beliebigen Zeichen beginnen, - Zahlen ausgenommen - und sie dürfen nicht mit Kernbefehlsnamen identisch sein. Dr. Logo beschwert sich, wenn Sie versuchen, den Namen eines Kernbefehls als Prozedurnamen zu verwenden (wenn Sie einen Kernbefehlsnamen trotzdem als Prozedurnamen verwenden möchten [siehe Anhang C, *Systemvariablen*], setzen Sie die Systemvariable REDEFP durch Eingabe des Ausdrucks make "REDEFP "TRUE auf TRUE).

Wenn Sie eine Zeile mit to beginnen, gibt Dr. Logo das Promptzeichen > aus, um zu zeigen, daß er die jetzt zur Definition der neuen Prozedur einzugebenden Anweisungen nicht sofort ausführen wird.

```
?to dreieck
>repeat 3 [fd 20 rt 120]
>end
dreieck defined
?
```

Das Schlüsselwort end zeigt Dr. Logo, daß Sie die Definition der Prozedur beendet haben. Dr. Logo gibt daraufhin das Prompt ? aus. Am Anfang der letzten Zeile einer Prozedur müssen Sie end allein eingeben. Jetzt, da dreieck definiert ist, können Sie

dreieck als Prozedurnamen verwenden. Wenn Sie einen Prozedurnamen auf das Prompt ? eingeben, wird Dr. Logo die entsprechende Prozedur wie folgt ausführen:

```
?dreieck
?
```

Verwenden Sie folgende Ausdrücke zur Beschreibung des Verhältnisses zwischen Namen und Definition einer Prozedur:

- o to dreieck ist die Titelzeile von dreieck
- o dreieck ist der Name von [repeat 3 [fd 60 rt 120]]
- o [repeat 3 [fd 60 rt 120]] ist die Definition von dreieck
- o [repeat 3 [fd 60 rt 120]] ist der Körper von dreieck

Der Körper einer Prozedur ist eine besondere Listenart: eine Liste von Ausdrücken. Diese Liste kann so viele Ausdrücke enthalten, wie Sie wollen, oder sie kann eine leere Liste sein. Da Dr. Logo die Definition einer Prozedur wie eine Liste behandelt, gibt es auch einen anderen Weg des Definierens von Prozeduren: das define-Kommando, das die eingegebene Definitionsliste ohne Anwendung der Schlüsselworte to und end zur Definition der angegebenen Prozedur macht (siehe Beispiel für define in Kapitel 8.).

Sie können Dr. Logos Text-Editor verwenden, um den Körper einer Prozedur zu ändern, eine neue Prozedur hinzuzufügen oder die Titelzeile einer bereits vorhandenen Prozedur zu ändern. Wenn Sie den Editor verlassen, sind die neuen oder veränderten Prozeduren definiert. Diese Definitionen ersetzen alle früheren Definitionen dieser Namen. Die Beschreibungen der Kernbefehle ed und edall in Kapitel 7 geben an, wie man Prozeduren in den Dr.-Logo-Editor laden kann, Kapitel 3 zeigt, wie man den Editor benutzt.

## 4.2 Schreiben übersichtlicher Prozeduren

Zwei traditionelle Methoden, die ein Programm übersichtlich machen, sind kurze Prozeduren und informative Prozeduren- und Variablennamen. Kurze Prozeduren sind im allgemeinen übersichtlicher, als lange. Kurze Prozeduren kann man auch leichter verstehen, testen und mit anderen Prozeduren kombinieren. Informative Prozedur- und Variablennamen helfen bei Beschreibung des Zwecks und der Wirkungsweise der Prozeduren oder der Art von Daten, die durch Variablen vertreten werden. Lange informative Namen können allerdings viel Speicherplatz besetzen, so daß Sie Ihre Prozedur- oder Variablennamen möglicherweise abkürzen müssen, wenn Sie ein langes Programm schreiben, das den ganzen Speicherbereich benötigt.

Eine andere Methode des Schreibens von lesbaren Prozeduren ist Strukturieren bzw. Formatieren der Prozeduren durch Anwendung von Einrückungen und Kom-

mentaren. Sie können eine lange Kommandozeile in mehrere Zeilen aufteilen und einige Zeilen dabei einrücken, um die Programmstruktur zu veranschaulichen, und Sie können Kommentare zur Beschreibung der Funktion der Prozedur verwenden.

### 4.2.1 Einrückungen

Eine Zeile in einer Prozedur ist im allgemeinen ein einziger Ausdruck: ein Prozedurname oder Kernbefehl und seine Eingabeparameter. Einige Prozeduren oder Kernbefehle erwarten allerdings komplizierte Eingaben, wie Listen von Anweisungen oder bedingte Ausdrücke. Solche Zeilen können daher so lang werden, daß ihre Übersichtlichkeit verlorengeht. Ein if-Kommando mit einem langen bedingten Ausdruck und zwei Anweisungen kann zum Beispiel die Breite Ihres Monitors leicht überschreiten. Wenn das geschieht, gibt Dr. Logo ein Ausrufezeichen aus und drückt den Rest des Kommandos in der nächsten Zeile:

```
?to Prüfe.die.Lieblinge
>if memberp :Süssigkeit.Fach :Lieblinge [pr [Essbare!
  Süssigkeiten verfügbar]] [pr [Vergiss es!]]
>end
Prüfe.die.Lieblinge defined
?
```

Sie können das Erscheinungsbild Ihrer Prozedur verbessern, indem Sie den langen Ausdruck in mehrere Zeilen aufteilen. Wenn Sie die Return-Taste drücken und die nächste Zeile mit einem Leerzeichen oder Tabulator beginnen, betrachtet Dr. Logo die neue Zeile als Fortsetzung der vorhergehenden Zeile. Dieser Vorgang wird Formatieren Ihrer Prozedur genannt.

```
?to Prüfe.die.Lieblinge
>if memberp :Süssigkeit.Fach :Lieblinge
> [pr [Essbare Süssigkeiten verfügbar]]
> [pr [Vergiss es!]]
>end
Prüfe.die.Lieblinge defined
?
```

### 4.2.2 Kommentare

Dr. Logo erlaubt Ihnen, Kommentare in Ihre Prozeduren einzufügen. Ein Kommentar ist ein Text, den Dr. Logo weder auszuwerten, noch auszuführen versucht. Benutzen Sie Kommentare, um die Funktion oder den Zweck einer Prozedur oder eines Ausdruckes zu beschreiben. Einem Kommentar muß ein Semikolon vorangestellt werden und der Kommentar muß das letzte oder einzige Objekt in der Zeile sein, wie im nächsten Beispiel gezeigt wird:

```
?to dreieck
>;Zeichne ein gleichschenkliges Dreieck
>repeat 3 [fd 20 rt 120]
>end
dreieck defined
?
```

Formatieren und Einfügen von Kommentaren verbraucht zusätzlichen Speicherplatz. Wenn dies zum Problem wird, können Sie den Kernbefehl `noformat` benutzen, um die Einrückungen und Kommentare aus Ihrem Arbeitsspeicherbereich zu entfernen.

### 4.3 Verwendung von Variablen in Prozeduren

Variablen eröffnen besondere Möglichkeiten innerhalb einer Prozedur. In diesem Abschnitt wird erläutert, wie man Ausdrücke unabhängig von den in ihnen manipulierten Daten schreibt, wie Sie Variablen zum Definieren der Eingabeparameter einer Prozedur verwenden können und wie man Informationen zwischen Prozeduren übergibt.

Es ist keine gute Programmiermethode, Konstanten in Ihren Prozeduren zu vergraben. In der folgenden Prozedur sind die Konstanten 40, 3, 20 und 120 enthalten:

```
?to flagge
>fd 40
>repeat 3 [fd 20 rt 120]
>bk 40
>end
flagge defined
?
```

Jemand, der Ihre Prozedur liest, kann mit einer Konstanten nicht viel anfangen. Erkennen Sie etwa, wenn Sie obige Prozedur lesen, gleich daß `fd 40` die Stange der Flagge zeichnet? Eine Konstante in einer Prozedur führt auch zu einer anderen Schwierigkeit: sie kann nicht schnell verändert werden.

Wenn Sie statt einer Konstanten eine Variable verwenden, sagt der Variablenname jemandem, der Ihre Prozedur liest, etwas darüber, was Ihre Prozedur macht: Sie können die Prozedur `flagge` zum Beispiel folgendermaßen verändern:

```
?to flagge
>fd :stange
>repeat 3 [fd :seite rt 120]
>bk :stange; Zurück zum Startpunkt
>end
flagge defined
?
```

Um die Ergebnisse der Prozedur `flagge` zu modifizieren, brauchen Sie vor dem Starten von `flagge` nur den Wert von `stange` und `seite` zu verändern. Die Prozedur `flagge` selbst brauchen Sie dabei nicht zu editieren. Die Ausdrücke der Prozedur sind von den Daten, die in ihnen manipuliert werden, unabhängig. So zum Beispiel:

```
?make "stange 50
?make "seite 30
?flagge
?
```

#### 4.3.1 Definieren der Eingabeparameter

Sie können eine Prozedur mit einer oder mehreren Variablen in der Titelzeile definieren, um so eine Prozedur zu erzeugen, die zur Ausführung Eingabe(n) erwartet. So zum Beispiel:

```
?to flagge :stange :seite
>fd :stange
>repeat 3 [fd :seite rt 120]
>bk :stange
>end
flagge defined
?
```

Eine Prozedur kann eine beliebig festgelegte Anzahl von Eingabeparametern haben. Wir haben die Prozedur `flagge` so modifiziert, daß sie jetzt zwei Eingabeparameter, `stange` und `seite` erwartet. Sie können die Werte dieser Parameter während der Ausführung der Prozedur direkt oder indirekt festlegen. Unten ist ein Beispiel für direkte Variablenwertzuweisung:

```
?flagge 40 20
?
```

Der Wert 40 wird der Variablen `stange` und der Wert 20 der Variablen `seite` während der Ausführung des Programms zugewiesen.

Ein Beispiel für indirekte Variablenwertzuweisung ist:

```
?make "gross 60
?make "klein 40
?flagge :gross :klein
?
```

Der Variablen `gross` wurde hier der Wert 60 und der Variablen `klein` der Wert 40 zugewiesen. Danach findet die Prozedur `flagge` die Werte der Variablen `gross` und `klein` und weist diese Werte den Variablen `stange` und `seite` zu.

Der Eingabeparameter einer Prozedur ist nur ein Objekt und der Variablenname ist sein Behälter. Der Prozess, den die Prozedur mit dem Eingabeparameter durchführt, bestimmt die Art des Eingabeparameters, der von der Prozedur erwartet wird. Die meisten Prozeduren erwarten eine bestimmte Art von Eingabeparametern. Eine Prozedur kann tatsächlich bis zum Erreichen des Moments normal funktionieren, wo sie auf eine Eingabe stößt, deren Art von der Art der erwarteten Eingabe abweicht. Wenn eine Prozedur zum Beispiel eine Zahl als Eingabe erwartet, wird sie mit einem Wort nicht viel anfangen können; und Dr. Logo wird sich beschweren:

```
?flagge "vierzig "zwanzig
fd doesnt't like vierzig in flagge: fd :stange
?
```

#### 4.3.2 Informationsübergabe zwischen Prozeduren

In bestimmten Fällen kann es notwendig werden, Werte von einer Prozedur zu anderen zu übergeben. Eine Prozedur kann den Wert einer Variablen benötigen, der während der Ausführung einer anderen Prozedur errechnet wurde, oder eine Prozedur kann den Ausgabewert einer anderen Prozedur benötigen.

Wir können die Prozedur `flagge` so weiter modifizieren, daß sie die Prozedur `dreieck` benutzt:

```
?to flagge :stange :seite
>fd :stange
>dreieck
>bk :stange
>end
flagge defined
?

?to dreieck
>repeat 3 [fd :seite rt 120]
>end
dreieck defined
?

?flagge 40 20
?
```

Beachten Sie besonders, wie die Prozedur `flagge` den ihr von Ihnen während der Laufzeit eingegebenen Wert von `seite` an die Prozedur `dreieck` übergibt.

Im folgenden Beispiel wird ein spezieller Fall einer Prozedur gezeigt, die den von der anderen Prozedur ausgegebenen Wert als Eingabeparameter benötigt. Die erste Prozedur, `durchschnitt`, ruft die Prozedur `aufadd` auf, um die Summe der Eingabezahlen von `durchschnitt` zu bestimmen. Um das Addieren durchzuführen, ruft `aufadd` sich selbst auf. Logo-Programmierer nennen eine Prozedur, die sich selbst aufruft, eine *rekursive Prozedur*.

Wenn Sie `durchschnitt` aufrufen, wird `durchschnitt` nur einmal ausgeführt, `aufadd` wird allerdings je einmal für jede Eingabezahl in der Eingabeliste von `durchschnitt` und noch einmal, wenn die Eingabeliste geleert ist.

```
?to durchschnitt :zahlen
>make "wieviele count :zahlen
>make "gesamt aufadd :zahlen
>op :gesamt / ;wieviele
>end
durchschnitt defined
?

?to aufadd :zahlen
>if empty? :zahlen [op 0]
>op (aufadd bf :zahlen) +
>(first :zahlen)
>end
aufadd defined
?

?durchschnitt [2 4 6]
4
?
```

Sie können die `trace`-Überwachung vor dem Aufruf der Prozedur `durchschnitt` einschalten. `trace` zeigt Ihnen, wie viele Male `aufadd` ausgeführt wird und den Wert der Variablen, die während der Programmausführung geändert werden. Wenn Sie fertig sind, geben Sie `notrace` ein.

## 4.4 Variablenarten

Es gibt zwei Arten von Variablen: *globale* und *lokale* Variablen. Globale Variablen sind im Arbeitsspeicherbereich definiert, sie sind daher für jede aufrufende Prozedur verfügbar. Lokale Variablen sind nur für eine Prozedur und jede Unterprozedur

definiert, die die betreffende Prozedur aufruft. Bei Beendigung ihrer Ausführung löscht die definierende Prozedur alle lokalen Variablen. Daher stehen lokale Variablen nach Beendigung der Ausführung der sie bestimmenden Prozeduren nicht mehr zur Verfügung.

Immer wenn Dr. Logo eine Variable erzeugt, stellt er die Variable und den ihr zugewiesenen Wert entweder in eine öffentliche (globale) oder in eine private (lokale) Bibliothek. Eine Prozedur sucht den Wert einer Variablen zuerst in ihrer eigenen, privaten Bibliothek. Wenn sie den Wert in der eigenen Bibliothek nicht findet, sucht sie in der privaten Bibliothek der Prozedur, die sie aufgerufen hat. Danach sucht sie in der Bibliothek der Prozedur, die aufrufende Prozedur aufgerufen hat, und so weiter. Am Ende, wenn sie den Namen der Variablen in keiner dieser privaten Bibliotheken findet, sucht sie in der öffentlichen Bibliothek der globalen Variablen.

Eine globale und eine lokale Variable können gleiche Namen haben, aber Dr. Logo setzt die beiden nicht gleich. Wenn der lokale Wert in einer privaten Bibliothek zur Verfügung steht, ignoriert Dr. Logo den Wert der globalen Variablen gleichen Namens.

#### 4.4.1 Erzeugen globaler Variablen

Dr. Logo verwendet eine spezielles Merkmal, um eine globale Variable zu erzeugen. In Kapitel 6 wird ausführlich erläutert, wie Sie die Merkmale zum Speichern und Abrufen der Information verwenden können, aber um zu verstehen, wie Dr. Logo die Merkmale zum Speichern und Abrufen globaler Variablen verwendet, müssen Sie zuerst verstehen, was *Merkmallisten*, *Merkmalepaare*, *Merkmalenamen* und *Merkmalewerte* sind.

Jedes beliebige Objekt im Arbeitsspeicherbereich kann eine Merkmalliste besitzen. Eine Merkmalliste besteht aus Merkmalepaaren. Das erste Element eines Merkmalepaars ist der Merkmalname, das zweite Element ist der Merkmalwert. Der Ausdruck `make` weist ein Merkmalepaar einem Variablennamen zu. So weist zum Beispiel der Ausdruck

```
?make "grösse 30
?
```

der Variablen `grösse` den Wert `30` zu, indem das Merkmalepaar `.APV` und `30` in die Merkmalliste von `grösse` eingetragen wird. Sie können den Kernbefehl `plist` zum Ausgeben der Merkmalliste dieser neuen Variablen verwenden:

```
?plist "grösse
[.APV 30]
?
```

In diesem Beispiel ist `[.APV 30]` die Merkmalliste von `grösse`. Sie enthält nur ein Merkmalepaar, obwohl sie viel mehr enthalten könnte, wie es in Kapitel 6 beschrieben wird.

`.APV` ist ein spezieller Merkmalname, der nur in der Merkmalliste einer globalen Variablen vorkommt. Dr. Logo erzeugt eine globale Variable durch Zuweisung des Variablennamens und des Merkmalepaars, bestehend aus `.APV` und dem Wert. `APV` steht für *Associated Property Value* (zugewiesener Merkmalwert), der Punkt vor `APV` bedeutet, daß es um ein Systemmerkmal handelt (die System-Merkmalnamen und ihre Funktionen sind in der Tabelle 6-1 und im Anhang D aufgeführt).

Um eine globale Variable zu erzeugen, können Sie einen `make`- oder `pprop`-Ausdruck verwenden. Unabhängig davon, ob Sie einen dieser Ausdrücke in der Hauptebene (Prompt: `?`) verwenden, oder von einer Prozedur aus aufrufen, Sie erzeugen damit eine globale Variable, und ihr Wert wird im Arbeitsspeicherbereich für jede Prozedur zur Verfügung stehen.

Ein `make`-Ausdruck, wie zum Beispiel

```
?make "Mädchen "Ursula
?
```

weist den Wert `Ursula` der Variablen `Mädchen` zu, indem er das Merkmalepaar `.APV` und `Ursula` in die Merkmalliste von `Mädchen` einträgt.

```
?plist "Mädchen
[.APV Ursula]
?
```

Der `pprop`-Ausdruck

```
?pprop "Junge ".APV "Johann
?
```

weist den Wert `Johann` der Variablen `Junge` zu, indem er das Merkmalepaar `.APV` und `Johann` in die Merkmalliste von `Junge` einträgt.

```
?plist "Junge
[.APV Johann]
?
```

#### 4.4.2 Erzeugen lokaler Variablen

Eine lokale Variable ist eine *temporäre* Variable, die von einer Prozedur für eigene Benutzung oder Benutzung durch die von ihr aufgerufenen Unterprozeduren erzeugt wird. Nach Beendigung der Ausführung der Prozedur, die die lokale Variable erzeugt

hatte, löscht Dr. Logo die lokalen Variablen aus dem Arbeitsspeicherbereich. Lokale Variablen haben keine Merkmalisten. Wenn Sie versuchen, einer lokalen Variablen ein Merkmal zuzuweisen, erzeugt Dr. Logo in Wirklichkeit eine globale Variable gleichen Namens mit dem eingegebenen Merkmalpaar.

Dr. Logo erzeugt eine lokale Variable, wenn Sie einen `local`-Ausdruck in einer Prozedur verwenden, oder wenn Sie eine Prozedur mit einem Eingabeparameter versehen. Wenn Sie zum Beispiel den `local`-Ausdruck

```
>local "Länge
```

in der Definition einer Prozedur verwenden, erzeugt Dr. Logo einen leeren Behälter namens `Länge`. Der nächste `make`- oder `pprop`-Ausdruck weist dann der lokalen Variablen einen Wert zu.

```
?to quadrat
>local "Länge
>make "Länge 50
>repeat 4 [fd :Länge rt 90]
>end
quadrat defined
?
```

Jedesmal, wenn die Prozedur ausgeführt wird, erzeugt Dr. Logo die lokale bzw. temporäre Variable `Länge` mit dem Wert 50. Wenn die Prozedur beendet wird, löscht Dr. Logo `Länge` aus dem Arbeitsspeicherbereich.

```
?quadrat
?:Länge
Länge has no value
?
```

Im folgenden Beispiel wird eine Prozedur mit einem Eingabeparameter dargestellt. Das ist die zweite Methode der Erzeugung einer lokalen Variablen:

```
?to flagge :stange
>fd :stange
>dreieck
>bk :stange
>end
flagge defined
?
?flagge 40
?
```

Diese letzte Kommandozeile erzeugt die temporäre (lokale) Variable `stange`. `stange` ist `flagge`s Eingabeparameter. Während der Ausführung des Programms müssen Sie der Variablen `stange` entweder direkt oder indirekt einen Wert zuweisen. Das folgende Kommando weist zum Beispiel der Variablen `stange` direkt den Wert 40 zu:

```
?flagge 40
?
```

Ein Beispiel für indirekte Wertzuweisung:

```
?make "gross 60
?flagge :gross
?
```

Wenn `flagge` diesmal ausgeführt wird, durchsucht die Prozedur die Bibliothek der globalen Variablen, findet dort den Wert von `gross`, und übergibt diesen Wert der temporären (lokalen) Variablen `stange`. Wenn `flagge` beendet wird, wird `stange` gelöscht, aber `gross` verbleibt in der Bibliothek der globalen Variablen.

Für die richtige Ausführung einiger rekursiver Prozeduren (Prozeduren, die sich selbst aufrufen) sind `local`-Ausdrücke notwendig. Da ein `local`-Ausdruck eine Variable vor Dr. Logos Suchprozess versteckt, können Sie diesen Ausdruck in einer rekursiven Prozedur verwenden, wenn Sie nicht wollen, daß die Prozedur den Wert der betreffenden Variablen bei jedem rekursiven Aufruf verändert. So zum Beispiel:

```
?to dezimalbruch :n
>local "kehrwert
>make "kehrwert 1 / :n
>if :n = 1 [stop]
>dezimalbruch :n - 1
>(pr [1 / ] :n char 9 :kehrwert
>end
dezimalbruch defined
?dezimalbruch 10
1 / 2 0.5
1 / 3 0.3333333333333333
1 / 4 0.25
1 / 5 0.2
1 / 6 0.1666666666666667
1 / 7 0.142857142857143
1 / 8 0.125
1 / 9 0.1111111111111111
1 / 10 0.1
?
```

```
? :kehrwert
kehrwert has no value
?
```

Im obigen Beispiel dient der Ausdruck `local`, um ein Überschreiben des alten Wertes von `kehrwert` durch jeden neuen Wert zu verhindern. In dieser Prozedur werden eigentlich neun lokale Variablen, alle mit dem Namen `kehrwert` erzeugt. Der Wert jeder dieser Variablen wird erst dann ausgegeben, wenn alle neun Werte berechnet worden sind. Die neun Variablen werden dann, eine nach der anderen, gelöscht, sobald die Ausführung der jeweiligen Prozedur beendet wurde. Versuchen Sie diese Prozedur ohne den `local`-Ausdruck zu verwenden und beobachten Sie, welche Ergebnisse ausgegeben werden.

#### 4.5 Klassifikation der Prozeduren

Die meisten Prozeduren können entweder als Kommandos oder als Operationen klassifiziert werden. Ein Kommando prozedur leitet eine Aktion ein und gibt keinen Wert aus, der von einer anderen Prozedur als Eingabe verwendet werden könnte. Kommandos können zum Beispiel den Igel bewegen, Bilder zeichnen oder Texte ausgeben.

Beachten Sie, daß die `po`-Kernbefehle, die Dinge im Textfenster ausgeben, Kommandos sind, die Ergebnisse ausgeben, aber nicht als Operationen betrachtet werden.

Eine Operation gibt ein Argument (ein Wort, eine Zahl oder eine Liste) zurück, was geeignet ist, als Eingabe für eine andere Prozedur verwendet zu werden. Der letzte auswertbare Ausdruck in einer Operationsprozedur ist immer ein `op`-Ausdruck, der die Ausführung dieser Prozedur beendet. Sie können natürlich eine Prozedur schreiben, die eine Aktion einleitet, bevor sie ein Argument ausgibt. Es ist allerdings besser, Prozeduren zu schreiben, die entweder einfache Kommandos oder einfache Operationen sind.

Bedingte Ausdrücke sind *logische* Operationen, die `TRUE` oder `FALSE` ausgeben, wie zum Beispiel `and`, `not` und `or`. Die Namen anderer logischen Operationen enden mit einem dem Kleinbuchstabe `p`, der für Aussage steht, wie zum Beispiel

```
?numberp "zwei
FALSE
?numberp 2
TRUE
?
```

Andere logische Operatoren, die Sie in Aussagen verwenden können, sind `empty`, `memberp`, `listp` und `wordp`. Es ist empfehlenswert, den Namen einer von Ihnen geschriebenen Prozedur mit einem `p` zu beenden, wenn sie `TRUE` oder `FALSE` ausgibt, um zu zeigen, daß es sich um eine logische Operation handelt.

Arithmetische Operationen geben Zahlen aus. Sie werden durch Symbole, wie `+`, `-`, `*` oder `/`, statt Namen gekennzeichnet. Einige logische Operationen, Vergleichsoperatoren genannt, verwenden auch Symbole, wie `>`, `<` und `=`, statt Kernbefehlsnamen. Prozeduren, die Symbole statt Kernbefehlsnamen verwenden, können entweder vorangestellte oder eingebettete Operatoren sein. Eingebettete Operatorsymbole stehen zwischen den Eingabeparametern im Ausdruck: `3 + 2`. Vorangestellte Operatorsymbole werden den Eingabeparametern vorangestellt: `+ 3 2`.

```
?2 * 3
6
?> 3 2
TRUE
?
```

#### 4.6 Auswertung der Prozeduren

Der Interpretierer von Dr. Logo, zu erkennen am Prompt `?`, wertet je eine ganze Zeile aus, so wie Sie sie über die Tastatur eingeben. Die Prozeduren wertet Dr. Logo ebenfalls zeilenweise aus. Wenn ein Ausdruck innerhalb einer Prozedur mit dem Namen einer anderen Prozedur beginnt, muß Dr. Logo zuerst die aufgerufene Prozedur ausführen, bevor er zur Auswertung der nächsten Zeile in der aufrufenden Prozedur zurückkehren kann. Er muß sich auch merken, an welcher Stelle er die Ausführung der aufrufenden Prozedur unterbrochen hat, besonders in dem Fall, in dem die aufgerufene Prozedur noch eine weitere Prozedur aufruft.

Zu diesem Zweck benutzt Dr. Logo einen sogenannten *Stapel* (Keller, Stack). Der Stapel ist ein Speicherbereich, der vom Interpretierer einerseits zur Speicherung der Werte der lokalen Variablen und andererseits zur Verwaltung der gerade laufenden Prozedur verwendet wird. In der Hauptebene (auf der höchsten Programmebene) befindet sich nichts im Stapel. Während der Ausführung befindet sich eine Prozedur im Stapel, bis sie eine andere Prozedur aufruft, dann werden es zwei. Wenn die zweite Prozedur eine dritte aufruft, dann werden es drei Prozeduren im Stapel und so weiter. Die Anzahl der Prozeduren im Stapel wird *Ebenennummer* genannt. Die Testeinrichtungen `trace` und `watch` geben die Ebenennummern während der Ausführung der Prozeduren aus. Die Hauptebene besitzt die Ebenennummer 0.

Dr. Logo teilt einen Teil des Speichers dem Stapel zu. Dr. Logo verwendet diesen Teil des Speichers auch zur Speicherung der Werte der lokalen Variablen. Wenn eine Prozedur sich selbst (rekursiv) aufruft und die Ebenennummer zu groß wird, oder

wenn eine Prozedur zu viele lokale Variablen definiert, kann der Stapelspeicherplatz überlaufen. Wenn das vorkommt, gibt Dr. Logo eine Meldung aus und beendet die Ausführung der Prozedur (siehe Kapitel 6 "Handhabung Ihres Arbeitsspeicherbereiches").

Eine Prozedur, die nie von anderen Prozeduren aufgerufen wird, aber selbst andere Prozeduren aufruft, wird Hauptebene-Prozedur und die von ihr aufgerufenen Prozeduren werden Unterprozeduren genannt.

## 5 Textbildschirm und Grafikbildschirm

Die Igelgraphik ist eine der faszinierendsten Fähigkeiten von Dr. Logo. Er kann mit einem graphischen Cursor, Igel genannt, Ihren Anweisungen entsprechend Linien zeichnen. Dieses Kapitel zeigt nicht, wie Sie den Igel steuern können, hier wird erklärt, wie Sie die physikalischen Eigenschaften Ihres Bildschirms steuern können. Es wird beschrieben, welche Kombinationen aus Text und Graphik möglich sind und wie Sie die Farbe (nur CPC6128) Ihrer Zeichnungen einstellen können. Außerdem werden hier die Kernbefehle `sf` und `tf` erläutert, die Sie zur Anzeige des aktuellen Zustands der Text- und Graphik-Bildschirmattribute verwenden können.

### 5.1 Bildschirme, Zeichenflächen und Fenster

Dr. Logo benutzt Ihren Systembildschirm, um Graphik, Text oder Kombinationen aus Text und Graphik darzustellen. Die *graphische Fläche*, auf der sich der Igel bewegen kann, ist logisch gesehen größer, als Ihr Bildschirm. Der Teil der graphischen Oberfläche, der auf Ihrem Bildschirm dargestellt werden kann, wird *Zeichenfläche* genannt. Der Teil des Textpuffers, der auf dem Bildschirm dargestellt werden kann, wird *Textfenster* genannt.

Wenn Dr. Logo auch Texte auf der Zeichenfläche ausgibt, nennt man diese Kombination der Text- und Graphikausgabe *geteilter Bildschirm* oder *Teilbildschirmmodus*. Wenn der ganze Bildschirm ausschließlich der Graphikausgabe dient, nennt man dies *Graphikbildschirm-* oder *Vollgraphikbildschirm-Modus*. Wenn der ganze Bildschirm ausschließlich der Textausgabe dient, nennt man dies *Textbildschirm-Modus*.

Die Kombination der Text- und Graphikausgabe auf Ihrem Bildschirm können Sie mit Kernbefehlen oder Kontrollzeichenkommandos steuern.

### 5.2 Text- und Graphikausgabe

Dr. Logo unterstützt Kernbefehle und Kontrollzeichenkommandos, die das Textfenster und die graphische Zeichenfläche auf Ihrem Bildschirm kombinieren können. Die entsprechenden Kernbefehle und Kontrollzeichenkommandos sind in Tabelle 5-1 zusammengefaßt.

Tabelle 5-1: Text- und Graphikausgabe

Kontrollzeichen	Kernbefehl	Bildschirmanzeige
Control-/Alt-L	fs	Nur die graphische Zeichenfläche wird angezeigt, der ganze Bildschirm dient ausschließlich der Graphikausgabe.
Control-/Alt-S	ss	Der Bildschirm wird aufgeteilt, es wird ein Textfenster auf der graphischen Zeichenfläche eröffnet.
Control-/Alt-T	ts	Nur das Textfenster wird ausgegeben, der ganze Bildschirm dient ausschließlich der Textausgabe. Achtung: Den Befehl bzw. die Tastenkombination müssen Sie zweimal eingeben, damit die gewünschte Wirkung eintritt!

Wenn Sie sich auf der Ebene des Prompts ? befinden, bieten die Kontrollzeichenkommandos die schnellste Möglichkeit, zwischen den Bildschirmmodi umzuschalten. Wenn Sie innerhalb einer Prozedur zwischen den Bildschirmmodi umschalten möchten, verwenden Sie die Kernbefehle fs, ss oder ts.

### 5.3 Bildschirmauflösung

Dr. Logo wird auf den Schneider Computern mit unterschiedlicher Bildschirmauflösung betrieben. Im Textmodus stehen Ihnen beim CPC6128 in einer Zeile 40 Zeichen zur Verfügung und beim Joyce sind es 80 Zeichen. An diesen Grundeinstellungen kann nichts verändert werden. Sie haben lediglich mit dem setscrunch-Kommando die Möglichkeit, das Bildseitenverhältnis der Zeichenfläche einzustellen.

### 5.4 Das Textfenster

Eine Textkoordinatenliste enthält zwei Zahlen: die erste Zahl ist die Spalten- oder Zeichennummer (die x-Koordinate); die zweite ist die Zeilennummer (die y-Koordinate). Die möglichen Bereiche, in denen die Koordinaten liegen können, hängen vom Rechner ab. In Anhang H finden Sie weitere Informationen.

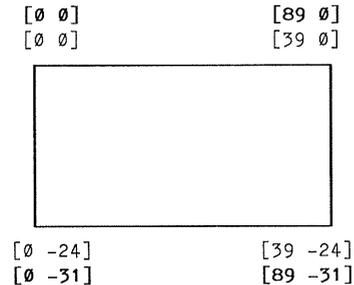


Abbildung 5-1: Textfensterkoordinaten beim CPC6128 und beim Joyce

Verwenden Sie obiges Koordinatensystem, wenn Sie die Position des Cursors im Textfenster mit setcursor einstellen. Verwenden Sie cursor wenn Sie die aktuelle Position des Cursors eingeben wollen.

```
?to diagtippen
>ct
>pr [Geben Sie eine Nachricht ein:]
>repeat 10
> [type rc
> setcursor list
> (3 + first cursor)
> (1 + last cursor)]
>end
diagtippen defined
?ts
?diagtippen
Geben Sie eine Nachricht ein:
```

```
H
  a
    l
      l
        o
          w
            e
              l
                t
?
?
```

### 5.4.1 Textkoordinaten auf geteiltem Bildschirm

Die Koordinaten, die zur Adressierung des Textfensters im Teilbildschirm-Modus verwendet werden, sind genau die gleichen, wie im Textbildschirm-Modus; es kann jedoch natürlich kein Text im Grafikbereich angezeigt werden.

Die Abmessungen des Textfensters bei geteiltem Bildschirm können mit dem `setsplit`-Kommando verändert werden.

## 5.5 Die Zeichenfläche

In Dr. Logo können Sie jeden einzelnen Punkt ansteuern, der zu Ihrer graphischen Zeichenfläche gehört. Beachten Sie, daß die Standardwerte des Bildseitenverhältnisses für alle Auflösungen so eingestellt sind, daß das Verhältnis zwischen der Punkt-zu-Punkt-Entfernung und dem Igelschritt ausgeglichen wird, d.h. daß ein Befehl, wie zum Beispiel `fd 50`, bei jeder Auflösungen das gleiche Bild ausgeben wird.

### 5.5.1 Die Koordinaten des Igels

Außer der Igelkommandos `fd`, `bk`, `lt` und `rt` können Sie jeden einzelnen Punkt auf der graphischen Zeichenfläche durch Angabe seiner Position einzeln ansteuern. Durch Verwendung der Koordinaten des betreffenden Punktes können Sie den Igel mit dem `setpos`-Kommando zu jeder Position schnell hinbewegen, mit dem `dot`-Kommando jedem einzelnen Punkt die aktuelle Igelfarbe geben oder mit dem `dotc`-Kommando (nur sinnvoll beim CPC6128) die Farbnummer des betreffenden Punktes ausgeben.

```
?to schnee
>make "x
> random 150 * (first shuffle [1 -1])
>make "y
> random 100 * (first shuffle [1 -1])
>dot list :x :y
>schnee
>end
schnee defined
?ht
?schnee (Esc bzw. Stop)
Stopped! in schnee: "x
?
```

Eine Koordinatenliste auf der graphischen Zeichenfläche besteht aus zwei Zahlen: aus der x-Koordinate, die die horizontale Position angibt, und aus der y-Koordinate,

die die vertikale Position bestimmt. Wenn Sie Dr. Logo zum ersten mal starten, ist die Ursprungsposition (`home`) des Igels `[0 0]`. Anders als beim Textfenster, in dem `ja [0 0]` die linke obere Ecke bedeutet, ist das jetzt der Mittelpunkt der graphischen Zeichenfläche. Um einen `dot` (Punkt) auf der Zeichenfläche anzugeben, kann man x-Koordinaten beim CPC6128 von -320 bis +319 und y-Koordinaten von -192 bis +191 eingeben. Bei dem Joyce reichen die x-Koordinaten ebenfalls von -320 bis +319, die y-Koordinaten jedoch von -192 bis +191

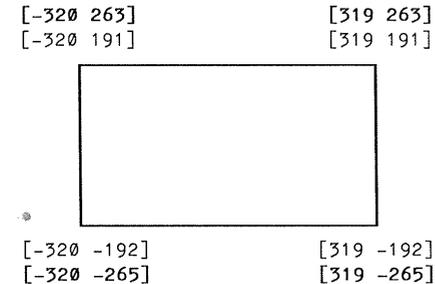


Abbildung 5-2: Koordinaten des Igels für den CPC6128 und den Joyce

Das `setpos`-Kommando liest die Koordinaten des Igels ein und bewegt den Igel auf der Zeichenfläche zur entsprechenden x-y-Position.

Die Koordinaten des Igels werden auch von einigen anderen Kernbefehlen benutzt:

<b>seth</b>	dreht den Igel so, daß er in eine bestimmte Richtung schaut.
<b>towards</b>	gibt die Richtung aus, in die sich der Igel drehen muß, um auf einen bestimmten Punkt zu zeigen.
<b>home</b>	Wenn Sie <code>home</code> eingeben, kehrt der Igel zur Anfangsposition <code>[0 0]</code> zurück und zeigt nach oben.
<b>clean</b>	löscht alle Zeichnungen von der Zeichenfläche ohne die Position des Igels zu verändern.
<b>cs</b>	löscht alle Zeichnungen von der Zeichenfläche und bringt den Igel in die <code>home</code> -Position.

Beachten Sie, daß die Zeichenfläche, ähnlich wie eine Luke in einem Raumschiff, nur ein Fenster in einer größeren graphischen Oberfläche ist. Sie können Zahlen außerhalb des sichtbaren Koordinatenbereichs angeben, um Positionen außerhalb der Zeichenfläche anzusprechen. Der Igel kann einen Teil Ihrer Zeichnung außerhalb des Bildschirms zeichnen und auf den Bildschirm zurückkehren, aber Sie können diesen Teil der Zeichnung nicht ansehen.

Sie können die Tätigkeit des Igels mit Hilfe von zwei Kernbefehlen, `fence` und `wrap` auf die Zeichenfläche begrenzen.

**fence** setzt eine Grenze um den Rand der Zeichenfläche. Wenn der Igel an diese Grenze stößt, wird sich Dr. Logo beschweren: `Turtle out of bounds` und er stoppt die Ausführung jeder Prozedur.

**wrap** läßt den Igel auf dem entgegengesetzten Rand der Zeichenfläche wiedererscheinen, wenn er die Grenze überschreitet. Zum Entfernen der Grenze können Sie `window` verwenden.

### 5.5.2 Textkoordinaten auf der graphischen Zeichenfläche

Auf der graphischen Zeichenfläche werden normalerweise graphische Daten und im Textfenster werden Textdaten ausgegeben. Sie können bei der Implementation von Dr. Logo auf den Schneider-Computern keine Texte auf der graphischen Zeichenfläche ausgeben.

### 5.5.3 Graphische Farben

Die mit Dr. Logo erstellten Igelgrafiken unterstützen die Farbfähigkeiten des CPC-Computers. Es gibt zwei verschiedene Möglichkeiten, die Farben zu steuern. Sie können die logischen Farben für den Hintergrund und für den Stift des Igels auswählen wie auch die physikalischen Farben, in denen die logischen Farben dargestellt werden. Beispiel: Sie wählen die Farbe 2 für den Stift und wählen dann blau für die logische Farbe 2. Der Stift und alle vorhandenen Punkte, Linien und Flächen mit Farbe 2 werden dann blau.

Die entsprechenden Kommandos zum Setzen der logischen Farbnummern für den Hintergrund und für den Stift sind `setbg` bzw. `setpc`. Beide akzeptieren eine Ziffer als Eingabe. Die Anzahl (entspricht der Anzahl der gleichzeitig anzeigbaren Farben) kann maximal 3 sein.

Die Beziehung zwischen den logischen Farbziffern und den tatsächlichen Farben kann mit dem Dr.-Logo-Kommando `setpal` geändert werden. Ansonsten ist er so eingestellt, daß `setbg 2` beispielweise immer die gleiche Farbe erzeugt.

Die Standardwerte für `setbg` und `setpc` sowie die Standardfarben für jede Farbziffer werden ebenfalls in Anhang H erläutert.

## 5.6 Bildschirmattribute

Es gibt zwei Kernbefehle, die Sie verwenden können, um den aktuellen Zustand des Bildschirms herauszufinden: `sf` und `tf`. Über die aktuellen physikalischen Eigenschaften der graphischen Zeichenfläche gibt `sf` Informationen aus und `tf` liefert Informationen über den aktuellen Zustand des Igels.

### 5.6.1 Attribute der graphischen Zeichenfläche

Der Befehl `sf` (screen facts) gibt eine Liste aus, die die aktuellen physikalischen Eigenschaften der graphischen Zeichenfläche enthält. Die Ausgabeliste von `sf` enthält 5 Werte: `HGFARBE`, `BILDMODUS`, `TEXTZEILEN`, `BERANDUNG` und `PROP`.

1. `HGFARBE` (`bgcolor_n`) gibt die aktuelle Nummer der Hintergrundfarbe an.
2. `BILDMODUS` (`screen state`) gibt `SS` (`split screen`) für geteilten Bildschirm, `TS` (`text screen`) für Textbildschirm und `FS` (`full screen`) für Vollgraphikbildschirm an.
3. `TEXTZEILEN` (`split size`) gibt die Anzahl der Textzeilen an, die auf der graphischen Zeichenfläche Platz haben.
4. `BERANDUNG` (`window-stae`) gibt `WINDOW`, `WRAP` oder `FENCE` an.
5. `PROP` (`scrunch`) gibt das aktuelle Bildseitenverhältnis der graphischen Zeichenfläche an.

### 5.6.2 Attribute des Igels

Mit `tf` (`turtle facts`) wird eine Liste ausgegeben, die die aktuellen Eigenschaften des Igels enthält. Die Ausgabeliste von `tf` besteht aus 6 Parametern: `XORT`, `YORT`, `KURS`, `STIFTZUSTAND`, `STIFTFARBE` und `IGELDA`.

1. `XORT` (`xcor`) zeigt die aktuelle x-Koordinate des Igels.
2. `YORT` (`ycor`) zeigt die aktuelle y-Koordinate des Igels.
3. `KURS` (`heading`) gibt die Richtung (in Winkelgraden) an, in die der Igel zeigt.
4. `STIFTZUSTAND` (`penstate`) zeigt an, ob der Farbstift des Igels angehoben (`SH`) oder abgesenkt (`SA`) ist, und ob er sich in Lösch-Modus (`RA`) oder Umkehr-Modus (`US`) befindet.

5. STIFTFARBE (pencocor\_n) zeigt die Nummer der aktuellen Stiftfarbe an.
6. IGELDA ist TRUE, wenn der Igel angezeigt wird, und FALSE, wenn er versteckt ist.

## 6 Merkmallisten und Speicherverwaltung

Speicherung ist eins der wichtigsten Hilfsmittel von Dr. Logo. Es gibt zwei Arten der Speicherung: *temporäre* und *permanente Speicherung*. Wenn Sie Definitionen von Prozeduren und Variablen über die Tastatur eingeben, werden diese in einem Teil des Speichers Ihres Computers, dem *Arbeitsspeicherbereich*, abgelegt. Um diese Definitionen permanent aufzuzeichnen, müssen Sie sie auf einer Diskette in Dateien abspeichern.

In diesem Kapitel wird erläutert, wie Sie Merkmallisten verwenden können, um Ihren Arbeitsspeicherbereich besser zu organisieren.

### 6.1 Merkmallisten

Jedes Objekt im Arbeitsspeicherbereich kann eine *Merkmalliste* besitzen. Dr. Logo verwendet Merkmallisten zur Erzeugung von Variablen und Prozeduren.

Eine Merkmalliste besteht aus *Merkmalpaaren*. Das erste Element eines Merkmalpaars ist der *Merkmalname*, das zweite Element ist der *Merkmalwert*. Zum Beispiel erzeugt Dr. Logo eine Variable durch Zuweisung des Merkmalpaars bestehend aus `.APV` und einem Wert zum Variablennamen.

**.APV** ist der Merkmalname (Associated Property Value = zugewiesener Merkmalwert). Das nächste Element in der Merkmalliste ist der zugewiesene Wert der Variablen.

Dr. Logo erzeugt eine Prozedur ähnlicherweise, durch Zuweisung des Merkmalpaars bestehend aus `.DEF` und einem Wert zum Prozedurnamen.

**.DEF** ist der Merkmalname (define = Definition). Der nächste Wert in der Merkmalliste ist die Anweisungsliste.

In Tabelle 6-1 sind Dr. Logos Systemmerkmale aufgeführt, die automatisch hinzugefügt oder entfernt werden, wenn Sie den Kernbefehl in der mittleren Spalte eingeben.

Tabelle 6-1: Merkmalpaare

Merkmalname	Kernbefehl	Merkmalwert
<code>.APV</code>	<code>make, ern, erall,</code>	Zugewiesener Merkmalwert. Wert einer globalen Variablen

<code>.DEF</code>	<code>to, define,</code> <code>er, erall,</code>	Definition einer Prozedur
<code>.ENL</code>	<code>to, ed</code>	Ende einer Prozedurzeile, die durch einen Wagenrücklauf und Leerzeichen oder Tabulatoren unterbrochen wurde.
<code>.FMT</code>	<code>to, ed</code>	Anfang einer Prozedurzeile, die durch einen Wagenrücklauf und Leerzeichen oder Tabulatoren unterbrochen wurde.
<code>.PRM</code>		Zeiger auf den Speicherplatz des Kernbefehls.
<code>.REM</code>		Bemerkung nach einem Semikolon.

Sie können diese Systemmerkmale mit dem `pprop`-Kommando manuell zuweisen. So weisen zum Beispiel beide folgenden Kommandozeilen der Variablen namens "Geschmack" den Wert "Schokolade" zu:

```
?pprop "Geschmack" ".APV" "Schokolade"
?make "Geschmack" "Schokolade"
?
```

Die folgende Kommandozeile weist der Prozedur "sage.hallo" den Wert [pr hallo] zu, genau so, als ob Sie die Prozedur durch Anwendung von `to` und `end` mit dem Editor oder mit dem `define`-Kommando definiert hätten.

```
?pprop "sage.hallo" ".define [[]] [pr "hallo"]
?
```

Das Format der Definitionsliste sieht genau so aus, wie die Eingabe, die für das `define`-Kommando erwartet wird (siehe Kapitel 8).

Verwenden Sie niemals Systemmerkmalnamen für andere Zwecke, als es in der Tabelle angegeben ist.

Einem Argument können Sie auch eigene (nicht systemeigene) Merkmale zuweisen. Merkmale eines Argumentes können verschiedene Beschreibungen dieses Arguments sein. Die Merkmale eines Buches könnten zum Beispiel `Autor`, `Art` (Roman oder Fachbuch) und die Anzahl der `Seiten` sein. Nicht-systemeigene Merkmale und ihre Werte können Sie, ähnlich wie unten, der Merkmalliste eines Arguments zuweisen:

```
?pprop "Unter.Geiern" "Autor" "May"
?pprop "Unter.Geiern" "Art" "Roman"
?pprop "Unter.Geiern" "Seiten" "544"
?
```

Die Merkmalliste eines Arguments können Sie mit dem `plist`-Kommando ausgeben:

```
?plist "Unter.Geiern"
[Autor May Art Roman Seiten 544]
?
```

Den Wert eines einzelnen Merkmals können Sie mit dem `gprop`-Kommando ausgeben:

```
?gprop "Unter.Geiern" "Autor"
Twain
?
```

Sie können beliebige Merkmalpaare erzeugen, die zu Ihrem Anwendungsfall passen. Merkmallisten erleichtern Abspeichern und Abrufen der Information. Sie können zum Beispiel eine Liste der von Ihnen am häufigsten benötigten Telefonnummern erzeugen:

```
?pprop "Robert" "Telefon" "089_4362123"
?pprop "Maria" "Telefon" "089_5682764"
?pprop "Ella" "Telefon" "089_2995368"
?
```

## 6.2 Verwaltung des Arbeitsspeicherbereiches

Der Arbeitsspeicherbereich ist ein Teil des Hauptspeichers im Computer. Dr. Logo verwendet diesen Speicherteil zur temporären Speicherung Ihrer Prozeduren und Variablen. In diesem Abschnitt wird gezeigt, wie Dr. Logo Ihren Arbeitsspeicherbereich zuteilt.

### 6.2.1 Die Größe des Arbeitsspeicherbereiches

Dr. Logo mißt Ihren Arbeitsspeicherbereich in sogenannten *Knoten*. Ein Knoten entspricht vier Bytes. Je mehr Arbeitsspeicherbereich Sie besitzen, desto besser ist im allgemeinen die Leistung Ihrer Prozeduren.

Die Anzahl der freien Knoten im Arbeitsspeicherbereich können Sie mit dem `nodes`-Kommando abfragen. Geben Sie `nodes` unmittelbar nach dem Start von Dr. Logo ein, um die maximale Größe Ihres Arbeitsspeicherbereiches zu erfahren. Beim CPC6128 sind es 3761 Knoten und beim Joyce 3643 Knoten.

Die Knoten werden nacheinander besetzt, so wie Sie Prozeduren und andere Objekte über die Tastatur eingeben oder mit `load` von einer Diskette in den Arbeitsspeicherbereich laden. Dr. Logo benutzt die Knoten auch zum Anfertigen von Kopien von

lokalen Variablen und rekursiven Prozeduren während der Ausführung der Prozeduren.

### 6.2.2 Speichersäuberung

Dr. Logo löscht die Kopien der lokalen Variablen und der rekursiv aufgerufenen Prozeduren nicht automatisch nach Beendigung der Ausführung der entsprechenden Prozedur. Wenn der freie Platz im Arbeitsspeicherbereich allerdings zu knapp wird, ruft Dr. Logo das Säuberungsprogramm auf, um den Arbeitsspeicherbereich durchzuforschten und die Kopien von nicht mehr benötigten Variablen oder Prozeduren zu löschen. Sie können gelegentlich sehen, daß die Ausführung einer Prozedur für kurze Zeit anscheinend unterbrochen wird, während das Säuberungsprogramm aufräumt.

Das Säuberungsprogramm können Sie mit dem `recycle`-Kommando auch manuell aufrufen. Wenn Sie vor dem Aufruf einer zeitkritischen Prozedur `recycle` aufrufen, vermindern Sie das Risiko, daß die Ausführung dieser Prozedur vom Säuberungsprogramm unterbrochen wird. Wenn Sie vor und nach dem `recycle`-Aufruf `nodes` aufrufen, sehen Sie, wie viele Knoten von vergessenen Kopien der Variablen und Prozeduren besetzt sind. Wenn Sie zum Entfernen der Prozedurformatierung (einschließlich Kommentare) `noformat` eingeben, müssen Sie auch `recycle` eingeben, damit das Säuberungsprogramm die Knoten wirklich freisetzt und den Arbeitsspeicherbereich zur besseren Ausnutzung reorganisiert.

Das Säuberungsprogramm benutzt den Stapel, wenn es die Objekte in Ihrem Arbeitsspeicherbereich neu ordnet. Der Stapel ist ein Speicherbereich, der vom Interpretierer zur Verwaltung der Werte der lokalen Variablen und zur Markierung der gerade laufenden Prozedur benutzt wird. Obwohl Dr. Logo den Stapelbereich dynamisch zuteilt, kann das Säuberungsprogramm oft nicht mehr genug Platz im Stapel finden, wenn es während der Ausführung einer rekursiven Prozedur oder während einer ähnlichen Situation aufgerufen wird, die den Stapel stark beansprucht. In diesem Fall gibt Dr. Logo eine Meldung aus und unterbricht die Ausführung der Prozedur.

Wenn der Arbeitsspeicher zu voll wird, gibt Dr. Logo die Meldung `Out of space` und ein Ausrufezeichen (!) als Prompt aus. Wenn das geschieht, geben Sie sofort `recycle` ein. Wenn das Säuberungsprogramm Ihre Prozeduren zu oft unterbricht, ist es höchste Zeit, daß Sie Ihre Prozeduren umorganisieren. Speichern Sie mit `save` einige Ihrer Prozeduren auf einer Diskette, um für die anderen Platz zu schaffen. Wenn Ihre Tätigkeiten den Arbeitsspeicherbereich überladen, können Sie Ihre Dateien auch innerhalb einer Prozedur laden und löschen. Diesen Vorgang nennt man *Dateiverkettung* (siehe Abschnitt 7.5.2).

## 7 Disketten, Laufwerke und Dateien

Beim Ausschalten Ihres Computers gehen alle Prozeduren und Variablen im Arbeitsspeicher verloren. Wenn Sie den Inhalt des Arbeitsspeichers sichern wollen, müssen Sie ihn vor dem Ausschalten des Computers auf einem dauerhafteren Medium, einer Diskette, abspeichern. Dr. Logo speichert die Information auf der Diskette in einer Datei, deren Namen Sie bestimmen können. Wenn Sie Ihre Dateien einmal auf einer Diskette abgespeichert haben, können Sie sie beim nächsten Einschalten wieder in den Speicher Ihres Computers laden.

Den Teil des Computers, der Dateien von einer Diskette liest und auf eine Diskette schreibt, nennt man Laufwerk. Die nächsten beiden Abschnitte enthalten Grundlageninformationen über Laufwerke und Disketten. Danach wird die Dateibehandlung erörtert.

### 7.1 Laufwerke

Zu Ihrem Computer gehören ein oder mehrere Laufwerke. Ein Laufwerk kann entweder ein- oder doppelseitig sein, d. h. das Laufwerk kann auf eine oder beide Seiten der Diskette schreiben (wie beim Joyce PCW8512).

Jedes Diskettenlaufwerk hat einen Namen, bestehend aus einem Buchstaben gefolgt von einem Doppelpunkt. In einem System mit nur einem Laufwerk heißt Ihr Laufwerk *A:*. In einem System mit zwei Laufwerken heißt das obere normalerweise *Laufwerk A:* und das untere *Laufwerk B:*.

Wenn Sie Dr. Logo vom Prompt `A>` aus starten, macht er Laufwerk A: zum angemeldeten Laufwerk. Das bedeutet, daß, solange Sie Dr. Logo nichts anderes vorschreiben, er die Informationen auf der Diskette in Laufwerk A: suchen wird. Wenn Sie Dr. Logo vom Prompt `B>` starten, macht er Laufwerk B: zum angemeldeten Laufwerk. Wenn Sie ein System mit nur einem Laufwerk besitzen, ist das Laufwerk A: normalerweise das angemeldete Laufwerk, es sei denn, Sie haben auch hier durch Eingabe von B: die Seite gewechselt. Sie können das angemeldete Laufwerk mit dem Kommando `setd` ändern. Der Kernbefehl `defaultd` gibt den Namen des angemeldeten Laufwerkes in Großbuchstaben aus.

```
?defaultd
A:
?setd "B:
?defaultd
B:
?
```

Wenn Sie beim Zugriff auf eine Datei die Laufwerkbezeichnung nicht angeben, nimmt Dr. Logo an, daß sich die Datei auf der Diskette im angemeldeten Laufwerk befindet.

Bevor Sie etwas auf Diskette abspeichern können, müssen Sie eine formatierte Diskette in ein Laufwerk legen. Wenn Sie ein Ein-Laufwerk-System besitzen, können Sie die Information auch auf der Dr.-Logo-Systemdiskette abspeichern. Sie können aber auch die Dr.-Logo-Systemdiskette aus dem Laufwerk entfernen und stattdessen eine Datendiskette einlegen.

## 7.2 Disketten

Dr. Logo kann keine Information auf unformatierte Disketten abspeichern. Die Diskette muß initialisiert oder ein- oder doppelseitig formatiert werden. vor. (Um Ihre Disketten zu formatieren, benutzen Sie das Programm DISCKIT3 bzw. DISCKIT.)

## 7.3 Dateien

Eine Datei ist eine Folge zusammenhängender Information, die auf einer Diskette gespeichert ist. Dr.-Logo-Dateien können Objekte, wie Prozeduren und Variablen mit ihren Merkmalslisten enthalten, oder ein Bild, das Sie gerade auf der graphischen Zeichenfläche gezeichnet haben. Eine Datei kann auch spezielle Daten, wie zum Beispiel Adressenlisten, enthalten.

Dr. Logo erstellt zwei verschiedene Arten von Dateien, abhängig davon, ob Sie Prozeduren oder Bilder speichern wollen. Dr. Logo gibt jeder Art von Dateien eine bestimmte Bezeichnung. Im nächsten Abschnitt wird die Bedeutung der Dateibezeichnung erklärt.

## 7.4 Dateibezeichnung

Dr. Logo legt eine Datei an, indem er ihre Bezeichnung in das Inhaltsverzeichnis der Diskette einträgt. *Dateibezeichnungen* sind wichtig, weil Dr. Logo die verschiedenen Arten von Dateien anhand der Bezeichnungen erkennt.

### 7.4.1 Was enthält eine Dateibezeichnung?

Eine Dateibezeichnung enthält Informationen, die Dr. Logo helfen, Ihre Datei zu finden. Die vollständige Dateibezeichnung besteht aus drei Teilen: dem *Diskettenlaufwerksnamen*, dem *Dateinamen* und aus dem *Dateityp*.

A:WERKZEUG.LOG

ist zum Beispiel eine vollständige Dateibezeichnung.

Der Diskettenlaufwerksname besteht aus einem Buchstaben gefolgt von einem Doppelpunkt: A: für das obere Laufwerk oder B: für das untere Laufwerk in einem Zwei-Laufwerk-System. Wenn Sie bei Bezugnahme auf eine Datei keinen Laufwerksnamen angeben, nimmt Dr. Logo an, daß sich Ihre Datei im angemeldeten Laufwerk befindet (Abschnitt 7.1 erklärt, wie man das angemeldete Laufwerk ändert.).

Die Datei WERKZEUG.LOG aus dem obigen Beispiel befindet sich auf der Diskette im Laufwerk A:. Sie könnten eine andere Datei mit dem Namen WERKZEUG.LOG auf der Diskette im Laufwerk B: besitzen. Der Laufwerksname sagt Dr. Logo, wo er eine Datei suchen muß.

Der Name einer Datei kann aus 1 bis 8 Zeichen bestehen. Wenn ein Dateiname aus mehr als 8 Zeichen besteht, werden von Dr. Logo nur die ersten 8 Zeichen benutzt. Auf die obige Datei wird normalerweise mit dem Namen WERKZEUG Bezug genommen.

Der Dateityp folgt dem Dateinamen. Der Dateityp besteht aus einem Punkt und aus 1 bis 3 Zeichen, die die Art der Datei anzeigen. Geben Sie keine Leerzeichen zwischen dem Dateinamen und dem Dateityp ein.

Bei der Wahl des Dateinamens und des Dateityps können Sie alle alphabetischen und numerischen Zeichen und folgende Sonderzeichen benutzen:

\$ & # ! % ' ' @ { | } \_ β †

Andere Zeichen haben in Dr. Logo besondere Bedeutung.

### 7.4.2 Bezugnahme auf mehrere Dateien

Dr. Logo kann auf mehr als eine Datei auf einmal Bezug nehmen, wenn Sie Ersatzzeichen im Dateinamen oder im Dateityp benutzen. Das Ersatzzeichen ist ein Fragezeichen, "?". Wenn Sie irgendein Zeichen im Dateinamen durch ein Fragezeichen, "?" ersetzen, wird Dr. Logo alle Dateien ansprechen, in denen die anderen Zeichen der eingegebenen Dateibezeichnung entsprechen und an der Stelle des Fragezeichens ein beliebiges Zeichen steht.

Beispiel: dir ist ein Kernbefehl, der die Namen der Dateien einer Diskette ausgibt:

```
?dir
[FIGUREN FARBEN TUCH]
?dir "F???????"
[FIGUREN FARBEN]
?
```

Die 7 Fragezeichen in der Kommandozeile sagen Dr. Logo, daß er alle mit dem Buchstaben F beginnenden Dateinamen mit einer Länge von bis zu 8 Zeichen ausgeben soll.

Eine Dateibezeichnung, die Ersatzzeichen enthält, wird manchmal *mehrdeutiger Dateiname* genannt.

Die Kernbefehle `dir`, `dirpic`, `erasefile` und `erasepic` können mehrdeutige Dateinamen einlesen. Seien Sie vorsichtig, wenn Sie Ersatzzeichen bei Kernbefehlen wie `erasefile` benutzen. Das Kommando

```
?erasefile "?????????.???"
?
```

würde alle `.LOG`-Dateien auf Ihrer Diskette löschen!

### 7.4.3 Dr. Logos Dateitypen

Wenn Sie bei Anwendung eines der Dateibehandlungs-Kernbefehle keinen Dateityp angeben, setzt Dr. Logo automatisch einen ein. Die vier Kernbefehle von Dr. Logo, die Ihnen das Erzeugen von Dateien ermöglichen, sind: `save`, `edf` und `savepic`. Die beiden Dateitypen von Dr. Logo sind:

```
.LOG      für Logo-Dateien
.PIC      für Bilddateien
```

Tabelle 7-1 zeigt die Dateitypen, die Dr. Logo während der Erzeugung der Datei zu dem von Ihnen eingegebenen Dateinamen automatisch hinzufügt. In der gleichen Tabelle sind Inhalt und Anwendungsmöglichkeiten der verschiedenen Dateiarten aufgeführt.

Tabelle 7-1: Dr. Logos Dateitypen

Erzeugender Kernbefehl	Vorgegebene Erweiterung	Dateityp, Inhalt und Verwendung
<code>save</code>	<code>.LOG</code>	Enthält Prozedur- und Variablendefinitionen und Merkmale, die im Arbeitsspeicher benutzt werden.
<code>edf</code>	<code>.LOG</code>	Enthält Prozedur- und Variablendefinitionen, Merkmale und einzelne Befehle, die im Arbeitsspeicher benutzt werden.
<code>savepic</code>	<code>.PIC</code>	Enthält graphische Entwürfe, die nicht im Arbeitsspeicher benutzt werden.

Wenn Sie eine Datei mit dem Kommando `save` erzeugen und keinen Laufwerksnamen oder Dateityp angeben, fügt Dr. Logo den Namen des angemeldeten Laufwerkes und den Dateityp `.LOG` zu Ihrem Dateinamen hinzu. Andere Kernbefehle der Datei-

behandlung funktionieren auf dieselbe Art und Weise, indem sie dem Dateinamen einen Laufwerksnamen und Dateityp hinzufügen. Sie können die vorgegebenen Werte nicht unterdrücken, indem Sie andere Werte angeben, wie zum Beispiel

```
?save "b:bilder.pic"
?
```

In den folgenden Abschnitten werden die beiden Dateitypen, die drei Methoden der Dateierzeugung und die verschiedenen Kernbefehle der Dateibehandlung erörtert, die Sie zur Verwaltung Ihrer Dateien benutzen können.

## 7.5 Logo-Dateien

Dateien mit dem Dateityp `.LOG` sind Logo-Dateien, die Ihre Dr.-Logo-Prozeduren, Variablen, und Merkmallisten enthalten. Die Kernbefehle der Dateibehandlung, die den Dateityp `.LOG` automatisch einsetzen, sind `changeif`, `dir`, `edf`, `erasefile`, `load` und `save`.

```
changeif  ändert den Namen einer Disketten-Datei. (Dieser Befehl funktioniert nicht in jedem Fall. Siehe auch in Kapitel 8 unter changeif)
dir       gibt eine Liste der Logo-Dateien auf Diskette aus.
edf       erzeugt oder modifiziert eine Datei direkt auf der Diskette.
erasefile löscht eine Datei von der Diskette.
load      lädt eine Datei von der Diskette in den Arbeitsspeicher.
save      speichert eine Datei vom Arbeitsspeicher auf die Diskette.
```

Um auf eine gewöhnliche Logo-Datei mit einem dieser Kernbefehle Bezug nehmen zu können, müssen Sie den Dateityp `.LOG` nicht angeben.

### 7.5.1 Anwendung von `save` zur Erzeugung einer Logo-Datei

Die gebräuchlichste Art der Dateierzeugung in Dr. Logo ist die Anwendung des Kommandos `save`. Wenn Sie `save` benutzen, schreibt Dr. Logo Information aus dem Arbeitsspeicher auf die Diskette. Und zwar entweder alles, was sich im Arbeitsspeicher befindet.

Eine Datei, die mit `save` erzeugt wurde, enthält verschiedene Objekte, wie Prozeduren und Variablen mit ihren Merkmallisten. Es ist eine gute Idee, Ihren Arbeitsspeicher zu *säubern*, bevor Sie Informationen auf eine Datei sichern wollen, so daß Sie keinen wertvollen Platz auf Ihrer Diskette mit unnötigen Daten besetzen. Die folgenden Kommandos helfen Ihnen, Ihren Arbeitsspeicher zu säubern:

```
ern      Das Kommando ern löscht die Namen oder Variablen, die von den Kommandos make oder pprop erzeugt wurden.
```

<b>er</b>	Das Kommando <code>er</code> löscht nicht erwünschte Prozeduren aus dem Arbeitsspeicher.
<b>remprop</b>	Das Kommando <code>remprop</code> löscht Merkmale, die einer Prozedur oder Variablen zugeordnet waren.

Bevor Sie Variablen und Prozeduren im Arbeitsspeicher löschen können, müssen Sie alle nicht systemeigenen Merkmale entfernen, die diesen Objekten zugeordnet sind (Kapitel 6 enthält genauere Informationen über die Verwaltung Ihres Arbeitsspeichers.).

Um eine Liste aller mit `save` erzeugten Dateien auszugeben, verwenden Sie das Kommando `dir`:

```
?dir
[WERKZEUG FIGUREN DSCHNITT]
```

### 7.5.2 Anwendung von `edf` zur Erzeugung einer Logo-Datei

Wenn Sie mit dem Kommando `edf` eine Datei erzeugen oder eine bereits existierende Datei modifizieren, aktualisiert Dr. Logo die Diskette nicht mit dem Inhalt des Arbeitsspeichers, wie beim Kommando `save`, sondern mit dem Editorpufferinhalt. Ein wesentlicher Vorteil des direkten Abspeicherns der Information aus dem Puffer des Texteditors, ist die Möglichkeit, einzelne Kommandozeilen oder Ausdrücke in die Diskettendatei zu schreiben, die bei Anwendung des Kommandos `save` verlorengehen würden.

Wenn Sie eine Datei, die einzelne Ausdrücke enthält, mit `load` laden, führt Dr. Logo die Befehle in der Reihenfolge aus, in der er sie von der Diskette liest. Unter Dr. Logo können Sie Dateien verketteten, indem Sie in die letzte Zeile einer Datei einen `load`-Ausdruck schreiben.

Nehmen wir an, daß Sie ein großes Programm entwerfen und Zugriff auf mehrere Dateien brauchen. In diesem Fall können Sie mit `edf` Kommandozeilen hinzufügen, die die notwendigen Prozeduren ausführen, dann unnötige Daten aus dem Arbeitsspeicher löschen und am Ende die nächste Datei mit `load` in den Arbeitsspeicher laden.

### 7.5.3 Bilddateien

Dateien mit dem Dateityp `.PIC` sind Bilddateien, die ein exaktes Abbild Ihrer graphischen Zeichenfläche enthalten. Sie können eine Bilddatei dazu benutzen, ein Bild wiederherzustellen, ohne daß es vom Igel erneut gezeichnet werden muß. Die Kernbefehle, die automatisch den Dateityp `.PIC` einsetzen, sind: `dirpic`, `erasepic`, `loadpic` und `savepic`.

<b>dirpic</b>	gibt eine Liste aller Dateien des Typs <code>.PIC</code> aus.
<b>erasepic</b>	löscht eine Bilddatei von der Diskette.
<b>loadpic</b>	stellt ein Bild auf der graphischen Zeichenfläche wieder her.
<b>savepic</b>	speichert eine Bilddatei auf die Diskette.

Um Namen von Bilddateien zu verändern, können Sie `changeif` mit Angabe des Dateityps `.PIC` verwenden. Wenn Sie eine Bilddatei erzeugen wollen, müssen Sie das Kommando `savepic` benutzen. Der Kernbefehl `savepic` erzeugt Bilddateien, indem er den Inhalt der graphischen Zeichenfläche in die angegebene Diskettendatei schreibt. Er speichert *nicht* die Prozeduren und Variablen ab, die zum Zeichnen des Bildes notwendig sind, es schreibt einfach ein Abbild des Bildschirms in die Datei. Beim Wiederherstellen von Bildern werden deshalb die zum Zeichnen des Bildes notwendigen Prozeduren, Variablen und Merkmalslisten nicht geladen. Wenn Sie diese Informationen speichern oder wieder laden wollen, müssen Sie normale Logo-Dateien und die Kommandos `save` und `edf` verwenden.

Die Proportionen eines wiederhergestellten Bildes werden durch die aktuellen Werte von `homenicht` beeinflusst. Bilddateien sind genaue Ebenbilder der graphischen Zeichenfläche.

## 8 Wörterbuch

In diesem Teil des Buchs sind die Definitionen von Dr. Logos Kernbefehlen aufgeführt und es wird gezeigt, wie korrekte Ausdrücke eingegeben werden. Ein Ausdruck ist die Folge aus einem Kernbefehl und beliebigen notwendigen Eingabeparametern. Obwohl Dr. Logo sowohl Groß-, als auch Kleinbuchstaben erkennen kann, müssen Sie *alle* Kernbefehle in Kleinbuchstaben eingeben. Genau so wie im Schulungsteil, werden auch in diesem Teil alle Eingaben der Benutzers in der **fetten Schriftart** und die Antworten und Meldungen von Dr. Logo in der *feineren Schriftart* gedruckt.

### 8.1 Format der Kernbefehle

Ebenso wie z.B. ein englisches Wörterbuch Sondersymbole benutzt, um Ihnen die Aussprache der Worte zu zeigen, benutzt das Wörterbuch der Dr.-Logo-Sprache auch Sondersymbole, um Ihnen zu zeigen, wie Sie einen Ausdruck eingeben können. Im Wörterbuch werden alle Kernbefehle - auch in der Überschrift- in *kleinbuchstaben* gedruckt, gefolgt von der Syntax (symbolische Struktur oder Format) der Eingabeparameter, sofern welche benötigt werden. Das Format zeigt Ihnen, wie Sie die Eingabeparameter eingeben müssen. Jeder Kernbefehl wird in folgendem Format angegeben:

**kernbefehl** Eingabeparameter

Geben Sie die Formatzeile genau so ein, wie es gezeigt ist.

Ein Kernbefehl akzeptiert eine bestimmte Anzahl und Art von Eingabeparametern. Die Mengen- und Typensymbole der Eingabeparameter sind unten in einer Liste zusammengefaßt. Die Mengensymbole geben an, wie viele male Sie einen Eingabeparameter eingeben können. Die Typensymbole zeigen den Typ der Eingabeparameter an, die der Kernbefehl akzeptieren kann.

### 8.2 Mengensymbole

In diesem Wörterbuch werden folgende Mengensymbole verwendet:

#### Runde Klammern (...)

Zeigen eine variable Anzahl von Eingabeparametern an. Sie können mehr oder weniger Eingabeparameter angeben, als normalerweise notwendig, indem Sie den ganzen Ausdruck, einschließlich den Kernbefehl, zwischen Klammern stellen. Wenn

hinter dem Ausdruck mit Klammern kein anderer Ausdruck steht, brauchen Sie die abschließende Klammer nicht einzugeben.

#### Spitze Klammern <>

Zeigen optionale Eingabeparameter an.

#### Oder-Strich |

Zeigt alternative Eingabeparameter an. Name | Namensliste zum Beispiel bedeutet, daß Sie entweder einen Namen oder eine Liste von Namen eingeben können.

### 8.3 Typensymbole

In diesem Wörterbuch werden folgende Typensymbole verwendet:

#### Wort

Bedeutet Eingabe eines Zeichens, einer Zahl oder einer Folge von Zeichen oder Zahlen. Außerhalb einer Liste müssen Sie jedem Wort einen Doppelpunkt (:) oder ein Anführungszeichen (") voranstellen. Trennen Sie Worte und Zahlen von anderen Objekten in der Zeile durch Leerzeichen oder durch eins der folgenden Begrenzungszeichen:

^ [ ] ; ( ) = < > + - \* /

Wenn Sie eines von diesen Begrenzungszeichen als normales Zeichen verwenden möchten, stellen Sie ihm einen umgekehrten Schrägstrich (\) voran.

#### Name

Bedeutet Eingabe eines speziellen Wortentyps: einer Prozedur, einer Variablen, einer Datei, eines Merkmals oder eines Laufwerks. Wenn Sie einen Namen ohne vorangestelltes Anführungszeichen eingeben, nimmt Dr. Logo stets an, daß es sich um einen Prozedurnamen handelt, und versucht diese Prozedur auszuführen. Wenn ein bestimmter Namenstyp erwartet wird, erscheint ein Symbol, wie folgt:

Varname	Variablenname
Prozname	Prozedurenname
d:	Laufwerksname. Ein Buchstabe und ein Doppelpunkt, die zusammen vor einem Dateinamen stehen können, um ein vom Standardlaufwerk verschiedenes Laufwerk anzugeben.
Datname	Dateiname, bis zu 8 Zeichen
Merkmalname	Das erste Element eines Merkmalpaares in einer Merkmalliste

#### n (Nummer)

Eingabe einer Zahl. n steht für eine beliebige Zahl; n hinter einem beschreibenden Ausdruck bedeutet eine spezielle Zahlenart. Grad\_n steht zum Beispiel für eine Anzahl von Winkelgraden.

#### Eingebettete oder vorangestellte Eingaben: a, b

Bedeutet entweder Eingabe eines eingebetteten Kernbefehls, wobei der Kernbefehl ein Symbol zwischen den Eingaben darstellt, oder Eingabe eines vorangestellten Kernbefehls, wobei der Kernbefehl vor den Eingaben steht. Hier sind a und b Objekte (Zahlen, Worte oder Listen). In arithmetischen Ausdrücken stehen a und b für Zahlen, wie zum Beispiel  $1 + 2$  oder  $12/6$ . In logischen Ausdrücken sind a und b beliebige Objekte, wie in  $[a \ b \ c] = [a \ b \ c]$ ,  $3 > 1$ , oder `:sohn = "robert`. Beachten Sie, daß *kleiner als* (<) und *größer als* (>) keine Listen als Eingabeparameter haben können.

#### Liste

Bedeutet Eingabe einer Folge von Objekten (Worten, Zahlen oder Listen), die zwischen eckige Klammern, [ und ], eingeschlossen sind. Eine Liste, die den Wert einer Variablen enthält, können Sie mit Hilfe der Kernbefehle `list` oder `se` zusammensetzen. Einige spezielle Listenarten:

Befehlsliste	auszuführende Befehle
Koord_Liste	ein x-y-Paar, das eine Stelle auf dem Bildschirm bestimmt.
Namensliste	Namen von Variablen und/oder Prozeduren

#### Objekt

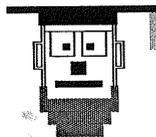
Eingabe eines beliebigen Objektes: eines Wortes (oder Namens), einer Zahl, einer Liste, oder eines Ausdrucks, der ein Wort, eine Zahl oder eine Liste ausgibt. Sie können auch Variablen statt beliebiger Objekte als Eingaben für eine Prozedur verwenden.

#### Aussage

Bedeutet Eingabe eines Ausdrucks, dessen Auswertung TRUE oder FALSE ausgibt.

### 8.4 Die Kernbefehle

Der Rest dieses Kapitels enthält eine Liste aller Dr.-Logo-Kernbefehle in alphabetischer Reihenfolge mit der Erklärung ihrer Funktion und mit Anwendungsbeispielen.



## A

**and** Aussage Aussage (...)

Gibt den Wert TRUE aus, wenn alle Aussagen *wahr* sind, andernfalls gibt er FALSE aus. Ein Beispiel finden Sie in der Prozedur *spirale* in der Definition von *co*.

```
?and (3<4) (7>3)
TRUE
?and "TRUE "FALSE
FALSE
```

**arctan** n

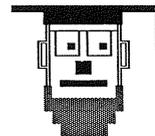
Gibt den Winkel (in Winkelgraden) zum eingegebenen Wert des Tangens aus.

```
?arctan 1
45
?arctan 10
84.2894068625003
```

**ascii** Wort

Gibt den ASCII-Wert des ersten Zeichens im eingegebenen Wort aus. Dieses Wort muß mindestens einen Buchstaben, eine Nummer oder ein Sonderzeichen enthalten.

```
?ascii "G
71
?ascii "g
103
?ascii "2
50
```



## B

**bf** Objekt

Gibt alle Elemente des eingegebenen Objektes mit Ausnahme des ersten Elementes aus.

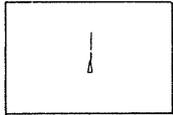
```
?bf "Opapa
papa
?bf [halb bittere Schokolade]
[bittere Schokolade]
```

**bk** Entfernung\_n

Bewegt den Igel um die als Eingabeparameter angegebene Anzahl von Schritten entgegen seiner aktuellen Richtung. Die aktuelle Richtung des Igels und der Zustand des Farbstiftes werden nicht geändert.



```
?cs
```



?bk 50

## bl Objekt

Gibt alle Elemente des eingegebenen Objektes mit Ausnahme des letzten Elementes aus.

```
?bl "Lehrer
```

```
Lehre
```

```
?bl [blöde Haselnuss]
```

```
[blöde]
```

```
?to verschlinge :objekt
```

```
>if empty? :objekt [stop]
```

```
>pr :objekt
```

```
>verschlinge bl :objekt
```

```
>end
```

```
verschlinge defined
```

```
?verschlinge "Huhn
```

```
Huhn
```

```
Huh
```

```
Hu
```

```
H
```

```
?
```

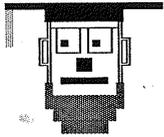
## buttonp Knopf\_n

Gibt den Wert TRUE aus, wenn der Knopf am angegebenen Steuerknüppel gedrückt ist; andernfalls gibt es FALSE aus. Die Knöpfe an den Steuerknüppeln sind durchnummeriert. Ein Beispiel der Anwendung von buttonp finden Sie bei paddle.

## bye

Beendigung der aktuellen Dr.-Logo-Sitzung. Alle Prozeduren und Variablen, die Sie noch nicht auf Diskette gesichert haben, gehen dabei verloren.

```
?bye
```



## C

**catch** Name Befehlsliste

Der Kernbefehl `catch` erkennt und behandelt Fehler und andere besondere Bedingungen, die während der Ausführung der eingegebenen Befehlsliste auftreten. Um besondere Bedingungen oder Fehler zu behandeln, können Sie einen `catch`-Ausdruck mit einem `throw`-Ausdruck verbinden. `catch` führt die eingegebene Befehlsliste solange aus, bis durch `throw` ein Fehler oder eine besondere Bedingung erkannt wird. Danach kehrt Dr. Logo zu `catch` zurück und führt die nächste Zeile aus. Ein solches *catch-throw-Paar* muß gleiche Eingabenamen verwenden. `TRUE` und `error` sind zwei spezielle Eingabenamen, die Sie anwenden können. `catch "TRUE"` erkennt jede `throw`-Ausführung und `catch "error"` erkennt jeden Fehler ohne Anwendung des `throw`-Ausdrucks. Beispiele von *catch-throw-Paaren* finden Sie auch in der Erklärung von `throw`.

```
?to meine.meldung
>catch "error [lese.befehl]
>(pr "AuWeh! first bf error [!])
>pr [Was wollen Sie machen?]
>run rl
>meine.meldung
>end
meine.meldung defined
```

```
?to lese.befehl
>pr [Was soll der Igel als nächstes tun?]
>run rl
>lese.befehl
>end
lese.befehl defined
```

```
?meine.meldung
Was soll der Igel als nächstes tun?
cs
```

```
Was soll der Igel als nächstes tun?
repeat 4 [df 50 rt 90]
Au Weh! I don't know how to df !
Was wollen Sie machen?
repeat 4 [fd 50 rt 90]
Was soll der Igel als nächstes tun?
ende
Au Weh! I don't know how to ende !
Was wollen Sie machen?
!Esc bzw. Stop!
Stopped! in meine.meldung: [run rl]
```

**change**f neuer\_Datname alter\_Datname

Ändert den Namen einer Datei im Inhaltsverzeichnis einer Diskette. Das `change`-Kommando überträgt die Datei *nicht* auf eine andere Diskette.

**Anmerkung:** Dieser Kernbefehl wurde zwar auf den Schneider-Computern bei der Anpassung von Dr. Logo implementiert, er funktioniert jedoch nicht. Eine Abhilfe wird vielleicht in absehbarer Zeit geschaffen. Inzwischen müssen Sie sich mit dem `REN`-Befehl des Betriebssystems behelfen, der in gleicher Weise wie `change` angewendet wird. Dazu müssen Sie allerdings Dr. Logo erst `bye` sagen und nach dem Umbenennen Dr. Logo wieder mit `logo3` bzw. `logo` starten.

```
?dir
[FARBEN FLIEGE FORMEN]
?change "tuch "farben
?dir
[TUCH FLIEGE FORMEN]
```

**char** n

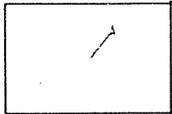
Gibt das zum eingegebenen ASCII-Wert (Zahl zwischen 0 und 255) gehörende Zeichen aus. Wenn Sie `char 8` (Rückschritt-Zeichen), `char 9` (Tabulatorzeichen), `char 10` (Zeilenvorschub) oder `char 13` (Wagenrücklauf) eingeben, führt Dr. Logo statt Ausgabe des Zeichens die entsprechende Funktion aus.

```
?char 103
g
?char 50
2
```

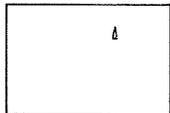
## clean

Löscht den Inhalt der Zeichenfläche, ohne die aktuelle Richtung und die aktuelle Position des Igel zu verändern. Vergleichen Sie `clean` mit `cs`.

```
?to treppe :Schritt
>repeat :Schritt
> [fd 5 rt 90
>   fd 5 lt 90]
>end
treppe defined
```



```
?treppe 9
```



```
?clean
```

## copyoff

Stoppt die Textausgabe auf dem Drucker, die mit dem `copyon`-Kommando gestartet wurde.

```
?copyoff
?dir
[TUCH FLIEGE FIGUREN]
```

*(Das wird nicht gedruckt)*

## copyon

Startet die Textausgabe auf dem Drucker. Die Druckausgabe kann mit `copyoff` gestoppt werden.

```
?copyon
?dir
?dir
[TUCH FLIEGE FIGUREN] (Das wird gedruckt)
```

## COS Grad\_n

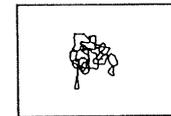
Gibt den Wert des Cosinus des eingegebenen Winkels (in Grad) in Form einer Dezimalzahl zwischen -1 und 1 aus.

```
?cos 0
1
```

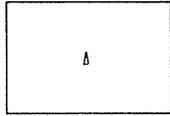
## CS

Löscht den Inhalt der Zeichenfläche und bewegt den Igel zur Ursprungsposition [0 0] zurück, der Farbstift wird abgesenkt. Vergleichen Sie `cs` mit `clean`.

```
?to krieche.zuf :anzahl
>repeat :anzahl
> [seth random 360 fd 7]
>end
krieche.zuf defined
```



```
?krieche.zuf 300
```



?cs

### CO <Objekt>

Setzt die Ausführung einer Prozedur fort, die durch pause, Control-Z oder einen Fehler (sofern ERRACT "TRUE" eingestellt ist) unterbrochen wurde. Geben Sie einen Eingabeparameter nach co nur dann ein, wenn Sie rc, rl oder rq mit Control-Z unterbrochen haben.

Anmerkung: Drücken Sie die Esc- bzw. Stop-Taste, um die unten angezeigte Prozedur spirale zu unterbrechen.

```
?to spirale :seite
>if and :seite > 80 :seite < 90 [pause]
>fd :seite rt 90
>spirale (:seite + 5)
>end
spirale defined
?spirale 20
pausing... in spirale: [if and :seite >80 :seite < 90 [pause]]
spirale ? :seite
85
spirale ?co
!Esc bzw. Stop!
Stopped! in spirale: spirale
?
```

### count Objekt

Gibt die Anzahl der Elemente im angegebenen Objekt aus. Das Objekt kann ein Wort, eine Zahl oder eine Liste sein. Um die Anzahl der Elemente eines Elements einer Liste zu bestimmen, geben Sie bitte einen item-Ausdruck als Eingabeparameter für count ein.

```
?count "Schokolade
9
?count [Vanille Erdbeer [Mokka Milch ]]3
?count item 3 [Vanille Erdbeer [Mokka Milch]]
2
```

### ct

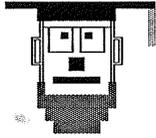
Löscht alle Texte im Textfenster, und positioniert den Cursor in die linke obere Ecke des Textfensters.

```
?pr [hallo!]
hallo?
ct
```

### CURSOR

Gibt die Koordinaten(-liste) [x y] aus, die die Spalten- und Zeilennummer der Cursorposition im Textfenster enthält.

```
?textschirm ct
?cursor
[0 1]
?(type [Die gegenwärtige Cursorposition ist] cursor)
Die gegenwärtige Cursorposition ist [36 3]
```



## D

### defaultd

Gibt den Namen des angemeldeten Laufwerkes aus.

```
?defaultd
A:
?setd "b:
?defaultd
B:
```

### define Prozname Definitionsliste

Legt die eingegebene Definitionsliste als Definition der Prozedur unter dem eingegebenen Prozedurnamen fest, ohne den Prozedureditor, to, oder end anzuwenden. Die Definitionsliste muß wiederum zwei Listen enthalten. Die erste Liste ist die Eingabeliste der Prozedur. Verwenden Sie dort keine Doppelpunkte vor den Eingabenamen. Wenn es keine Eingaben gibt, muß die erste Liste innerhalb der Definitionsliste leer sein. Die zweite Liste in der Definitionsliste ist die Definition der Prozedur. Die Definition muß als eine Textzeile ohne end-Anweisung eingegeben werden. Innerhalb der Definitionsliste der Prozedur dürfen sich noch weitere Listen befinden.

```
?define "hallo [[wer] [(pr "hallo, :wer)]
?po "hallo
to hallo :wer
(pr "hallo, :wer)
end
?hallo "David
hallo, David
```

### dir <Datname>

Gibt eine Liste der Dr.-Logo-Dateien auf der angemeldeten oder der angegebenen Diskette aus. dir akzeptiert auch mehrdeutige Dateinamen. Es wird die Liste der Dateien ausschließlich vom Dateityp .LOG ausgegeben. Wenn Sie eine Liste aller Bilddateien (Dateityp .PIC) erhalten wollen, müssen Sie den Kernbefehl dirpic anwenden. Wenn Sie eine Liste aller Dateien erhalten wollen, müssen Sie Dr. Logo mit bye verlassen und nach dem Betriebssystem-Prompt (A>) dir eingeben..

```
?dir
[FIGUREN TUCH FARBEN TELEFON]
?dir "B:
[SPULE FLIEGE.LOG]
?dir "t???????
[TUCH TELEFON]
```

### dirpic <Datname>

Gibt eine Liste der Bilddateien auf der angemeldeten oder angegebenen Diskette aus. Der Kernbefehl dirpic akzeptiert auch mehrdeutige Dateinamen. Die dirpic-Liste enthält nur Dateien mit dem Dateityp .PIC. savepic fügt automatisch den Dateityp .PIC zum Dateinamen einer Datei hinzu.

```
?savepic "zeichn1
?dirpic
[ZEICHN1 BLUME GESICHT]
?dirpic "B???????
[BLUME.PIC]
```

### dot Koord\_Liste

Setzt einen Punkt in der aktuellen Stifffarbe, dem aktuellen Farbstiftzustand entsprechend in die durch die eingegebene Koordinatenliste festgelegte Position. Der dot-Befehl ändert weder die Position, noch die Richtung oder die Farbe des Igels.

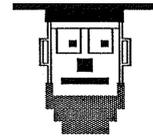
```
?dot [50 50]
```

## dotc

Gibt die Farbnummer in der durch die eingegebene Koordinatenliste bestimmten Position aus. Wenn diese Position außerhalb der Zeichenfläche liegt, gibt dotc den Wert -1 aus.

Der dotc-Befehl ist zwar auf dem Joyce anwendbar, sinnvoll einsetzen kann man ihn aber nur auf dem CPC6128.

```
?make "X 1
?cs st pd setpc 1
?repeat 30 [fd :X rt 90 make "X :X + 1]
?setpc 2
?repeat 30 [fd :X rt 90 make "X :X + 1]
?dotc [0 0]
1
?dotc [0 20]
2
```



## E

### ed Name | Namensliste

Lädt die angegebene(n) Prozedur(en) und/oder Variable(n) in den Texteditorpuffer. In Abschnitt 4.3 finden Sie die ausführlichere Beschreibung der Kontrollzeichenkommandos und Funktionstasten des Texteditors.

Wenn die Ausführung einer Prozedur aufgrund eines Fehlers abgebrochen wird, können Sie ed ohne Angabe des Prozedurnamens aufrufen. Dr. Logo lädt automatisch die fehlerhafte Prozedur in den Texteditor und positioniert den Cursor in die Zeile, in der der Fehler aufgetreten ist.

Eine neue Prozedur können Sie statt mit dem Prozedureditor auch mit dem Texteditor schreiben. Wenn Sie ed aufrufen und den Namen der neuen Prozedur angeben, fügt Dr. Logo die to- und end-Zeilen automatisch hinzu.

Um den Inhalt des Editorpuffers zu sichern, drücken Sie Copy (CPC6128) bzw. Exit (Joyce). Wenn Sie den Editorpuffer ohne Aktualisierung des Arbeitsspeichers verlassen wollen, drücken Sie Esc bzw. Stop. Sie können den verlassenen Editorpuffer mit dem ed-Kommando wieder abrufen und unverändert verwenden, wenn zwischen dem Abbruch mit Esc bzw. Stop und ed noch nichts eingegeben wurde.

#### ?quadrat

```
I don't know how to repaet in quadrat: repaet 4 [fd 25 lt 90]
?ed
```

```
to quadrat
repaet 4 [fd 25 lt 90]
end
```

## edall

Lädt alle Variablen und Prozeduren aus dem Arbeitsspeicherbereich in den Editorpuffer. Wenn der Arbeitsspeicher nichts enthält, wird auch der leere Puffer des Texteditors angezeigt, in den Sie dann Prozeduren oder Variablen eingeben können.

?edall

## edf Datname

Lädt die angegebene Datei in den Texteditorpuffer oder erzeugt eine Datei und ruft den Texteditor mit leerem Puffer auf. Wenn Sie den Texteditor mit Copy (CPC6128) bzw. Exit (Joyce) verlassen, aktualisiert Dr. Logo nur die Diskette und nicht den Arbeitsspeicherbereich. Wenn Sie Esc bzw. Stop drücken, wird die Editiersitzung unterbrochen und die Diskette wird von Dr. Logo nicht aktualisiert. Wenn Sie mehr über die Anwendung von edf wissen wollen, schlagen Sie bitte in Abschnitt 7.5.2 nach.

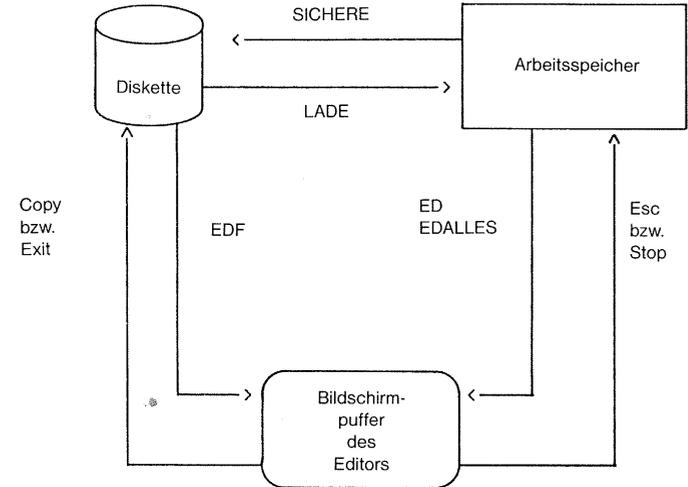


Abbildung 8-1: Anwendung von edf

## empty Objekt

Gibt den Wert TRUE aus, wenn das eingegebene Objekt ein leeres Wort oder eine leere Liste ist. Ansonsten wird FALSE ausgegeben. Wenn Sie feststellen wollen, ob bereits alle Elemente des eingegebenen Objektes einer Prozedur abgearbeitet sind, verwenden Sie empty als Aussage in einem if-Kommando.

```
?empty [Dies ist eine Liste von Wörtern]
FALSE
?empty[]
TRUE
?make "grössen []
?if empty :grössen [pr "HALT]
HALT
```

## end

Zeigt dem Prozedureditor, daß Sie die Definition der Prozedur abgeschlossen haben, und führt Sie zum Prompt ? zurück. Wenn Sie den Prozedureditor benutzen, setzen Sie `end` ohne weitere Ausdrücke in die letzte Zeile Ihrer Prozedurdefinition. Wenn Sie zur Definition einer Prozedur den Texteditor benutzen und dabei die `end`-Zeile vergessen, fügt Dr. Logo diese Zeile automatisch am Ende der Prozedur hinzu. In Abschnitt 4.2 finden Sie weitere Information über die Anwendung des Prozedureditors.

```
?to fünfack
>repeat 5 [fd 25 lt 72]
>end
fünfack defined
```

**ent\*** Hüllkurven\_n Schritt\_Zahl Schritt\_Per Schrittzeit

Mit dem `ent`-Befehl definieren Sie die Ton-Hüllkurve, die Sie über die Hüllkurven-Nummer im `sound`-Befehl abrufen können. Bis zu 16 *Tonhöenschwankungen*, von 0 bis 15, sind zugelassen.

**Hüllkrvn\_n** Über diese Nummer wird die gewünschte Hüllkurve im `sound`-Befehl abgerufen.

**Schritt\_Zahl** Mit der Schrittzahl bestimmen Sie, wie oft sich während einer Periode die Tonhöhe ändert. Um einen 5 Sekunden langen Ton je halbe Sekunde die Tonhöhe schwanken zu lassen, geben Sie die Schrittzahl 10 ein ( $5 / 0,5 = 10$ ). Die Schrittzahl kann Werte von 0 bis 239 annehmen.

**Schritt\_Per** Die Stärke der Tonhöenschwankung wird mit diesem Parameter festgelegt. Der nutzbare Bereich liegt zwischen -128 und 127, wobei negative Schritte die Tonhöhe ansteigen und positive Schritte die Tonhöhe abnehmen lassen. Die 0 erzeugt die kürzeste Schwankung.

**Schrittzeit** bestimmt die Zeit zwischen den einzelnen Tonhöenschwankungen. Zugelassen sind die Werte von 0 bis 255, wobei einem Schritt von 1 Sekunde der Wert 100 entspricht. Der Wert 0 wird als 256 interpretiert und gibt damit die Höchstdauer von 2,65 Sekunden an.

\* Nicht auf dem Joyce implementiert.

```
?ent [3 100 2 2]
?sound [1 478 50 15 1 3]
?
```

**env\*** Hüllkurven\_n Schritt\_Zahl Schritt\_Weite Schrittzeit

Mit dem `env`-Befehl stellen Sie die Lautstärkeschwankungen ein, die Sie über die Hüllkurven-Nummer im `sound`-Befehl abrufen können. Bis zu 16 Lautstärke-Hüllkurven, von 0 bis 15, sind zugelassen.

**Hüllkrvn\_n** Über diese Nummer wird die gewünschte Hüllkurve im `sound`-Befehl abgerufen.

**Schritt\_Zahl** Mit der Schrittzahl bestimmen Sie, wie oft sich während einer Periode die Lautstärke ändert. Um einen 5 Sekunden langen Ton je halbe Sekunde die Lautstärke schwanken zu lassen, geben Sie die Schrittzahl 10 ein ( $5 / 0,5 = 10$ ). Die Schrittzahl kann Werte von 0 bis 127 annehmen.

**Schritt\_Wte** Obwohl im `sound`-Befehl die Lautstärke nur zwischen 0 und 15 variiert werden kann, liegt der nutzbare Bereich des Schrittweit-Parameters zwischen -128 und 127. Dabei geht die Lautstärke wieder auf 0 zurück, wenn das Maximum erreicht ist.

**Schrittzeit** bestimmt die Zeit zwischen den einzelnen Lautstärkeschwankungen. Zugelassen sind die Werte von 0 bis 255, wobei einem Schritt von 1 Sekunde der Wert 100 entspricht. Der Wert 0 wird als 256 interpretiert und gibt damit die Höchstdauer von 2,65 Sekunden an.

```
?env [2 10 1 50]
?sound [1 284 500 1 2]
?
```

**er** Prozname | Prozname Liste

Löscht die Definitionen aller Prozeduren aus dem Arbeitsspeicherbereich. Dabei wird das `.DEF`-Paar aus der Merkmaliste der Prozedur gelöscht. Wenn die Prozedur ande-

\* Nicht auf dem Joyce implementiert.

re Merkmalpaare außer .DEF hat, müssen diese mit remprop entfernt werden, um den Prozedurnamen aus dem Arbeitsbereich zu löschen.

```
?plist "rad
[.APV 12.speichen .DEF [[] [repeat 12 [flagge lt 30]]]
?er "rad
?plist "rad
[.APV 12.speichen]
?remprop "rad ".APV
?plist "rad
[]
```

## erall

Löscht die Definitionen aller Prozeduren und Variablen aus dem Arbeitsspeicherbereich. Dabei wird die .DEF-Paare aus der Merkmalliste der Prozeduren gelöscht. Wenn die Prozeduren andere Merkmalpaare außer .DEF haben, müssen diese mit remprop entfernt werden, um sämtliche Prozedurnamen aus dem Arbeitsbereich zu löschen.

```
?erall
?to quadrat :grösse
>repeat 4 [fd :grösse rt 90]
>end
quadrat defined
?pots
to quadrat :grösse
?erall
?pots
?
```

## erasefile Datname

Löscht einen Dateinamen des Typs .LOG von der Diskette im angegebenen Laufwerk. Auch mehrdeutige Dateinamen werden akzeptiert. Um eine Bilddatei (Dateityp .PIC) zu löschen, müssen Sie den erasepic-Befehl anwenden.

```
?erasefile "b:farben
?load "farben
File farben not found
?
```

## erasepic Datname

Löscht die angegebene Bilddatei. Auch mehrdeutige Dateinamen werden akzeptiert.

```
?dirpic
[ZZEICHN1 ZEICHN2 ZEICHN3 BLUME GESICHT]
?savepic "b:quadrante
?dirpic "b:
[B:QUADRATE]
?erasepic "z????????
?dirpic
[BLUME GESICHT]
```

## ern Varname | Varnamensliste

Löscht die angegebene(n) Variable(n) aus dem Arbeitsspeicherbereich. Dabei wird die .APV-Paare aus der Merkmalliste gelöscht, die mit den Variablen verbunden sind. Wenn die Variable(n) andere Merkmalpaare außer .APV haben, müssen diese mit remprop entfernt werden, um sämtliche Prozedurnamen aus dem Arbeitsbereich zu löschen.

```
?make "uc 50
?make "mittel 40
?make "lc 30
?ern "mittel
?pons
lc is 30
uc is 50
```

## error

Gibt eine Liste bestehend aus sechs Elementen aus, die den zuletzt aufgetretenen Fehler beschreibt. Wenn error bereits eine Meldung im Textfenster ausgegeben hat, wird eine leere Liste ausgegeben. Die Fehlerliste enthält folgende Elemente:

1. Eine Fehlernummer. Die Tabelle der Dr.-Logo-Fehlermeldungen ist in Anhang E aufgeführt.
2. Eine Meldung, die den Fehler erläutert. Das ist die Meldung, die normalerweise im Textfenster ausgegeben wird.

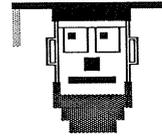
3. Der Name der Prozedur, die den fehlerhaften Ausdruck enthält. Wenn der Fehler in der Hauptebene aufgetreten ist, ist dieses Element eine leere Liste.
4. Die vollständige Zeile, die den fehlerhaften Ausdruck enthält.
5. Der Prozedurnamen-Teil des fehlerhaften Ausdrucks, sofern er existiert.
6. Der Objektteil des fehlerhaften Ausdrucks, sofern er existiert.

Wenn die Systemvariable `ERRACT` auf `TRUE` gesetzt wurde, und ein Fehler erkannt wird, gibt Dr. Logo eine `pausing...-Meldung` aus und unterbricht die Ausführung der Prozedur, anstatt sie zu stoppen. Wenn Sie die Beschreibung des Fehlers auf dem Bildschirm erhalten wollen, geben Sie `error` ein. Während der Pause können Sie den Editor benutzen, um Definitionen von Variablen oder die der Prozedur zu ändern. Sie können von hier, genau so, wie von der Hauptebene (Prompt `?`), auch andere Prozeduren aufrufen. Wenn Sie die Ausführung der Prozedur fortsetzen wollen, geben Sie `co` ein. Wenn Sie die Prozedur verlassen wollen, geben Sie `stop` ein.

```
?to sicherheits.kreis :grösse
>catch "error
> [repeat 180
>   [fd :grösse rt 2]]
>stop
>end
sicherheits.kreis defined
```

```
?sicherheits.kreis
Not enough inputs to sicherheits.kreis in sicherheits.kreis :!
sicherheits.kreis
```

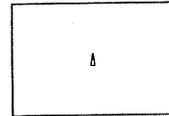
```
?make "ERRACT "TRUE
?sicherheits.kreis
pausing... in sicherheits.kreis:[sicherheits.kreis]sicherheits!
.kreis ?error
[29 [Not enough inputs to sicherheits.keis] sicherheits.kreis!
 [sicherheits.kreis] [] []]
sicherheits.kreis ?stop
```



## F

### fd Entfernung\_n

Bewegt den Igel um die eingegebene Zahl von Schritten in seiner aktuellen Richtung und hinterläßt eine Spur entsprechend dem Zustand und der Farbe des Farbstiftes.



```
?cs
```



```
?fd 25
```

**fence**

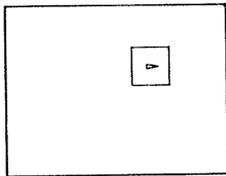
Ändert den Bildschirmmodus, wenn `window` oder `wrap` eingestellt ist. Erzeugt eine Grenze um die graphische Zeichenfläche, die den Igel am Zeichnen außerhalb der sichtbaren Zeichenfläche hindert. Um `fence` zu stornieren, geben Sie `window` oder `wrap` ein. Der `fence`-Modus kann nur dann eingeschaltet werden, wenn sich der Igel innerhalb der Grenzen der graphischen Zeichenfläche befindet.

```
?fence
?home fd 300
Turtle out of bounds
```

**fill\***

Füllt einen Zeichenbereich in der aktuellen Stifffarbe entsprechend dem aktuellen Zustand des Farbstiftes auf. Dabei werden der Bildpunkt unter dem Igel und alle Bildpunkte gleicher Farbe umgefärbt, die zusammen mit dem Bildpunkt unter dem Igel eine zusammenhängende Farbzone bilden. Die Wirkung von `fill` endet an jeder Grenze zu anderen Farbzonen und an der Grenze der graphischen Zeichenfläche.

```
?to füllequ
>repeat 4 [fd 50 rt 90]
>pu
>fd 25 rt 90 fd 25
>pd fill
>end
füllequ defined
```



```
?füllequ
```

\*Nur auf dem Joyce implementiert

**first** Objekt

Gibt das erste Element des eingegebenen Objektes aus.

```
?first "Rose
R
?first [Gewicht 165]
Gewicht
```

**fput** Objekt Objekt

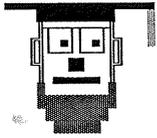
Gibt ein neues Objekt aus, indem es das erste Objekt als erstes Element zum zweiten Objekt hinzufügt. Mit `fput` kann man allerdings keine Liste in ein Wort integrieren.

```
?fput "B "und
Bund
?fput 1 [2 3]
[1 2 3]
```

**fs**

Schaltet den Bildschirmmodus aus den Zuständen `ss` oder `ts` um und stellt den ganzen Bildschirm für Graphikausgabe zur Verfügung. Im `fs`-Modus verschwindet der Textbereich. Sie können immer noch Kommandos eingeben, aber Ihre Eingabe ist auf dem Bildschirm unsichtbar. Wenn Dr. Logo eine Fehlermeldung ausgeben muß, schaltet er automatisch auf `ss`-Modus um, so daß die Fehlermeldung sichtbar wird.

```
?ss
?fs
?ss
?ts
```



## G

### glist Merkmal

Gibt eine Liste aller Objekte aus, die das angegebene Merkmal in ihrer Merkmalliste enthalten. Diese Objekte müssen sich im Arbeitsspeicher befinden.

```
?glist ".DEF
[abzählen farben anfang.selbstlaut fliege klingel schwein]
```

### go Wort

Führt die Zeile innerhalb der laufenden Prozedur aus, die einem label-Ausdruck gleichen Namens folgt. Die entsprechenden go- und label-Ausdrücke müssen gleiche Eingabeworte enthalten und sich in der gleichen Prozedur aber in verschiedenen Zeilen befinden. Es dürfen keine zusammengehörenden go- und label-Ausdrücke innerhalb einer Anweisungsliste der Befehle repeat, run oder if verwendet werden.

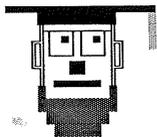
```
?to textdreieck :zeichenkette
>label "schleife
> if empty :zeichenkette [stop]
> pr :zeichenkette
> make "zeichenkette bf :zeichenkette
>go "schleife
>end
textdreieck defined
?textdreieck [Auf geht's, kleiner Hund]
Auf geht's, kleiner Hund
geht's, kleiner Hund
kleiner Hund
Hund
```

### gprop Objektname Merkmalname

Gibt den Wert des angegebenen Merkmals im angegebenen Objekt aus.

```
?gprop "schwein ".DEF
[[Wrt] if anfang.selbstlaut :Wrt [op wor!
d :Wrt "Ay] [op schwein word (bf :Wrt) (!
first :Wrt)]]]
```

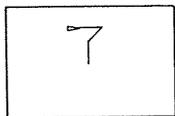
```
?make "Höhe "72"
?gprop "Höhe ".APV
72"
?gprop "Höhe ".DEF
[]
```



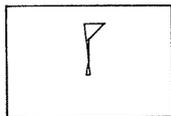
# H

## home

Setzt den Igel auf die Zeichenflächenkoordinaten [0 0], in Richtung Norden (nach oben). Der Kernbefehl `home` verändert den Zustand des Farbstiftes nicht. (Siehe `cs` und `clean`.)



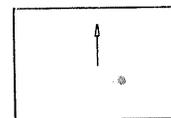
```
?fd 50 rt 45
?fd 50 rt 45
?bk 50
```



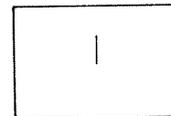
```
?home
```

## ht

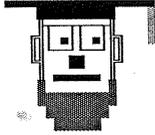
Macht den Igel unsichtbar. Wenn der Igel unsichtbar ist, zeichnet er schneller und hebt sich visuell von der Zeichnung nicht ab. Benutzen Sie `st`, um den Igel wieder sichtbar zu machen. (Anmerkung: Wenn Sie Farbrnummer 0 angewählt haben, sind Stiftfarbe und Hintergrundfarbe gleich, und der Igel scheint zu verschwinden. In diesem Fall können Sie den Igel mit `st` *nicht* wieder sichtbar machen, sondern Sie müssen `setpc` benutzen, um die Farbe des Farbstiftes zurückzusetzen.)



```
?cs fd 25
```



```
?ht
```



# I

## if Aussage Befehlsliste <Befehlsliste>

Abhängig davon, ob der Wert der angegebenen Aussage TRUE oder FALSE ist, wird die erste oder die zweite Befehlsliste ausgeführt. Benutzen sie `if`, um Entscheidungen bezüglich des Ablaufes innerhalb einer Prozedur zu treffen.

Die erste Eingabe zu `if` muß eine Aussage sein: ein Ausdruck, der TRUE oder FALSE ausgibt. Wenn die Aussage TRUE ist, führt Dr. Logo die erste Befehlsliste aus. Ist die Aussage FALSE, wird entweder die zweite Befehlsliste (sofern sie vorhanden ist), oder sofort die nächste Zeile der Prozedur ausgeführt.

Die Befehlslisten müssen *literal* sein, d. h. sie müssen in eckige Klammern eingeschlossen sein. Dr. Logo erlaubt *verschachtelte* `if`-Ausdrücke.

```
?to münze
>if 1 = random 2
>  [pr "Kopf]
>  [pr "Zahl]
>end
münze defined
?münze
Kopf
?
```

## int<sub>n</sub>

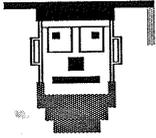
Gibt den Ganzzahlwert der eingegebenen Zahl aus, wobei das Dezimalkomma und alle nachfolgenden Stellen gelöscht werden. Der Eingabebereich von `int` geht von -32769 bis +32768.

```
?int 3.333333
3
?int 28753 / 12
2396
```

## item<sub>n</sub> Objekt

Gibt das `n`-te Element des angegebenen Objektes aus.

```
?item 4 "Gnom *
m
?item 2 [Capuchino Lutschbonbon Waffel]
Lutschbonbon
```

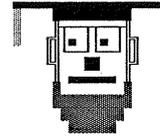


## K

### keyp

Gibt den Wert TRUE aus, wenn ein Zeichen über die Tastatur eingegeben wurde, das von rc, rq oder rl gelesen werden kann. Benutzen Sie keyp, um festzustellen, ob der Puffer die Eingabenart enthält, die Ihre Prozedur benötigt, bevor Sie die Eingabe an die Prozedur weitergeben.

```
?to taste.skizze
>fd 2
>wait 10
>if keyp [drehe rc]
>taste.skizze
>end
taste.skizze defined
?to drehe :towards
>if "r = :towards [rt 10] ;nach rechts drehen
>if "l = :towards [lt 10] ;nach links drehen
>end
drehe defined
?taste.skizze
!Esc bzw. Stop!
Stopped! in taste.skizze: [wait 10]
```



## L

### label Wort

Das label-Kommando identifiziert die Zeile, die nach dem go-Ausdruck mit derselben Bezeichnung ausgeführt werden soll. Die label- und go-Ausdrücke müssen in der gleichen Prozedur sein und die gleichen Worte benutzen.

```
?to countdown :n
>label "schleife
> if :n < 0 [stop]
> type :n
> make "n (:n - 1)
>go "loop
>end
countdown defined
?countdown 9
9876543210
?
```

### last Objekt

Gibt das letzte Element des angegebenen Objektes aus.

```
?last "Wolkenkratzer
r
?last [Gewicht 165]
165
```

**lc** Wort

Gibt das eingegebene Wort vollständig in Kleinbuchstaben aus.

```
?lc "OST
ost
?lc "FiBu
fibu
```

**list** Objekt Objekt (...)

Gibt eine in eckige Klammern eingefügte Liste aus, die aus den eingegebenen Objekten zusammengesetzt wurde. Vergleichen Sie `list` mit `se`. In der Beschreibung von `run` finden Sie ein Beispiel der Anwendung von `list` in einer Prozedur.

```
?show list "grosse "Füsse
[grosse Füsse]
?(list "A "B "C "D)
[A B C D]
?list "Kartoffeln [gekocht gebacken gebraten]
Kartoffeln [gekocht gebacken gebraten]
```

**listp** Objekt

Gibt den Wert `TRUE` aus, wenn das eingegebene Objekt eine Liste ist. Ansonsten wird `FALSE` ausgegeben.

```
?listp "Sternenstaub
FALSE
?listp [Erde Luft Feuer Wasser]
TRUE
```

**load** Datname

Liest die eingegebene Logo-Datei (Dateityp `.LOG`) von der Diskette in den Arbeitsspeicher ein.

Beim Lesen von der Diskette führt `load` jede Zeile der Datei aus, d. h. es definiert jede Prozedur und Variable so, als ob die Eingabe über Tastatur erfolgt wäre. `load`

führt auch jeden alleinstehenden Ausdruck aus, der mit `edf` zur Datei hinzugefügt wurde. Wenn Sie zum Beispiel die Zeile `pr "hallo` mit `edf` zu einer Datei hinzufügen, drückt `load` auf dem Textfenster beim Lesen der Datei `hallo` aus.

Wenn Sie Dateien verketten wollen, verwenden Sie einen `load`-Ausdruck als letzte Zeile einer Datei. Sie erzeugen damit eine Datei, die, wenn sie geladen wird, Prozeduren ausführt, Elemente aus dem Arbeitsspeicher löscht und eine andere Datei lädt.

Der Kernbefehl `load` kann nur mit `Esc` bzw. `Stop` abgebrochen werden, `pause` oder `Control-Z` sind zu diesem Zweck nicht geeignet.

```
?load "farben
anfang.selbstlaut defined
schwein defined
farben defined
```

**loadpic** Datname

Gibt die Zeichnung auf der Zeichenfläche aus, die in der eingegebenen Datei gespeichert ist. Der Befehl `loadpic` kann nur Dateien lesen, die mit `savepic` erzeugt wurden. Er definiert nicht die Prozeduren und Variablen, die zur Erstellung des Bildes notwendig sind.

```
?savepic "blume
?dirpic
[BLUME GESICHT]
?loadpic "blume
```

**local** Varname (...)

Stellt die angegebene(n) Variable(n) nur der aktuellen Prozedur und ihrer Unterprozedur(en) zur Verfügung. In Abschnitt 4.4.2 finden Sie in der Prozedur `dezimalbruch` ein Beispiel der Anwendung von `local`.

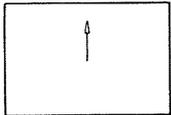
**lput** Objekt Objekt

Gibt ein neues Objekt aus, indem er das erste Objekt zum letzten Element des zweiten Objektes macht. Sie können jedoch mit `lput` zu einem Wort keine Liste hinzufügen.

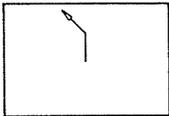
```
?lput "s "Auto
Autos
?lput "s [Auto]
[Auto s]
?lput "Kirschen [Apfel Bananen]
[Apfel Bananen Kirschen]
```

**lt** Grad\_n

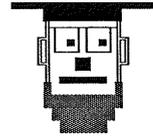
Dreht den Igel um den in Grad eingegebenen Winkel nach links (gegen den Uhrzeigersinn). Die Eingabe muß zwischen -32767 und +32767 liegen. Eine negative Zahl dreht den Igel nach rechts.



```
?fd 50
```



```
?lt 45 fd 40
```

**M****make** Varname Objekt

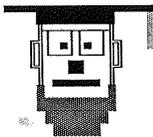
Macht das eingegebene Objekt zum *Inhalt* oder Wert der angegebenen Variablen. Der Kernbefehl `make` trägt das Systemmerkmal `.APV` mit dem Wert des Eingabeobjekts in die Merkmalliste der Variablen ein. In Abschnitt 6.1 finden Sie weitere Information über Dr. Logos Merkmale.

```
?make "Geschmack "Schokolade
?:Geschmack
Schokolade
?make "Schokolade "Halbbitter
?:Schokolade
Halbbitter
?thing "Geschmack
Schokolade
?thing :Geschmack
Halbbitter
```

**memberp** Objekt Objekt

Gibt den Wert `TRUE` aus, wenn das erste eingegebene Objekt ein Element des zweiten ist, ansonsten wird `FALSE` ausgegeben.

```
?memberp "n "nur
TRUE
?memberp 7 734395
TRUE
?memberp "Schokolade [[Vanille] [Schokolade] [Erdbeer]]
FALSE
?memberp [Schokolade] [[Vanille][Schokolade][Erdbeer]]
TRUE
```




---

## N

---

### namep Wort

Gibt den Wert `TRUE` aus, wenn das eingegebene Wort eine definierte Variable ist, ansonsten wird `FALSE` ausgegeben.

```
?make "Geschmack "Schokolade
?:Geschmack
Schokolade
?namep "Geschmack
TRUE
?namep "Schokolade
FALSE
```

### nodes

Gibt die Anzahl freier Knoten im Arbeitsspeicher an. Ein Knoten entspricht vier Bytes.

```
?nodes
2684
?recycle
?nodes
3030
```

### noformat

Entfernt Prozedurformatierung und Kommentare aus dem Arbeitsspeicher, um mehr freien Speicherplatz zu bekommen.

```
?nodes
2684
?recycle
?nodes
2790
?noformat
?recycle
?nodes
3030
```

### not Aussage

Kehrt den Wert einer Aussage um. Gibt den Wert `TRUE` aus, wenn die eingegebene Aussage `FALSE` war und `FALSE`, wenn die Aussage `TRUE` war.

```
?not 2 > 1
FALSE
?not 2 < 1
TRUE
?to kehrwert :n
> if not numberp :n
> [(pr :n [ist keine Zahl.]) stop]
>(pr "1 "/ :n "= 1 / :n
>end
kehrwert defined
?kehrwert 5
1 / 5 = 0.2
?kehrwert fünf
fünf ist keine Zahl.
?
```

### notrace

Stoppt die von `trace` eingeleitete Überwachung der Prozedurausführung und Variablenzuweisung. In der Erklärung von `trace` finden Sie ein Anwendungsbeispiel von `notrace`.

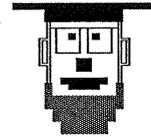
**nowatch** <Prozname | Proznamensliste>

Stellt die von `watch` eingeleitete Überwachung aller angegebenen Prozeduren ab. In der Erklärung von `watch` finden Sie ein Anwendungsbeispiel von `nowatch`.

**numberp** Objekt

Gibt den Wert `TRUE` aus, wenn das eingegebene Objekt eine Zahl ist, ansonsten wird `FALSE` ausgegeben. In der Erklärung von `not` finden Sie in `kehrwert` ein Anwendungsbeispiel von `numberp` in einer Prozedur.

```
?numberp 374.926
T                TRUE
?numberp "SECHS
FALSE
?numberp first [2 4 6 8]
TRUE
?numberp bf [2 4 6 8]
FALSE
```



O

**op** Objekt

Gibt das angegebene Objekt als Ergebnis der Prozedur aus. Sobald die Prozedur ein Objekt ausgibt, wird die Steuerung an die aufrufende Prozedur zurückgegeben oder das Hauptebene-Prompt erscheint. Der `op`-Befehl kann nur in Prozeduren verwendet werden.

```
?to selbstlaut :objekt
>op memberp :objekt [a e i o u ä ö ü]
>end
selbstlaut defined
?selbstlaut "r
FALSE
?selbstlaut "e
TRUE
?
```

**OR** Aussage Aussage (...)

Gibt den Wert `FALSE` aus, wenn alle eingegebenen Aussagen `FALSE` ergaben. Wenn mindestens eine einzige Aussage `TRUE` war, wird `TRUE` ausgegeben.

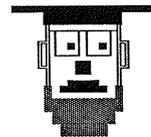
```
?or (1<3) (1=3)
TRUE
?(or (2*5=11) (6/2=4) (16<15))
FALSE
```

```
?to wetter
>type [Wie warm ist es heute?\ ]
>make "antwort rq
>if emptytyp :antwort
```

```

> [pr [Ich wünsche Dir einen schönen Tag.] stop]
> if or (:antwort < 10 ) (:antwort > 32)
> [pr [Hmm, klingt ungemütlich!]]
> [pr [Klingt gut!]]
>wetter
>end
wetter defined
?wetter
Wie warm ist es heute? 37
Hmm, klingt ungemütlich!
Wie warm ist es heute? 25
Klingt gut!
Wie warm ist es heute? !cr!
Ich wünsche Dir einen schönen Tag.
?

```




---

## P

---

### paddle\* n

Gibt eine der Koordinaten eines Steuerknüppels (Joystick) als Zahl aus.

```

?to zeichn
>rt (paddle 0) - 65
>fd paddle 1
>zeichn
>end
zeichn defined
?zeichn

```

### pal\* Farb\_n

Der pal-Befehl gibt die Liste der RGB-Nummern für die angegebene Farbnummer aus. Der Bereich der erlaubten Nummern für die Farben rot, grün und blau liegt zwischen 0 und 3. Mit setpal können Sie die Farbzuoordnung des mit setpc eingestellten Farbstifts ändern.

```

?setpal 3 [1 0 1]
?pal 3
[1 0 1]
?setpc 3
?show item 5 tf
3
?pal (item 5 tf)
[1 0 1]

```

---

\* Der Kernbefehl paddle ist nicht auf dem Joyce implementiert.  
 \* Nicht auf dem Joyce implementiert.

## pause

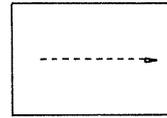
Unterbricht die Ausführung der gerade laufenden Prozedur und erlaubt Ihnen, mit dem Interpretierer oder dem Editor interaktiv zu arbeiten. Während einer pause können Sie Definitionen von Variablen oder Prozeduren verändern oder sogar andere Prozeduren laufen lassen, so als ob Sie sich beim Prompt ? befänden. Eine pause ist nur innerhalb einer Prozedur gültig. Um die unterbrochene Prozedur fortzusetzen, geben Sie `co` ein. Wenn Sie die Prozedur endgültig abbrechen wollen, geben Sie `stop` ein.

```
?to nautilus :grösse
>if (remainder :grösse 5) = 0 [pause]
>repeat 36 [fd :grösse rt 10]
>rt 15 nautilus :grösse + 0.5
>end
nautilus defined
?cs rt 180
?nautilus 1
pausing... in nautilus: [if (remainder :grösse 5 = 0 [pause])]
nautilus ? :grösse
5
nautilus ?co
pausing... in nautilus [if (remainder :grösse 5 = 0 [pause])]
nautilus ?stop
?
```

## pd

Senkt den Farbstift des Igels ab, um das Zeichnen nach `pu`, `pe` oder `px` in der aktuellen Farbe fortzusetzen. Beim Starten von Dr. Logo ist der Farbstift abgesenkt.

```
?to stift.ab :länge :anzahl
>pu setpos [-100 0]
>seth 90 pd
>repeat :anzahl
> [fd :länge pu
>   fd 2 pd]
>end
stift.ab defined
```



```
?stift.ab 30 10
```

## pe

Der Igel zeichnet in der Hintergrundfarbe, er löscht daher jede Zeichnung, die er überfährt. Um das Löschen zu beenden, benutzen Sie `pu` oder `pd`.

```
?to bewegl.dreieck
>cs
>ht
>repeat 36
> [dreieck
>   pe dreieck
>   pd setpc 2 rt 10]
>end
bewegl.dreieck defined
?to dreieck
>repeat 3 [fd 40 rt 120]
>end
dreieck defined
?bewegl.dreieck
```

**piece** n n Objekt

Gibt ein Objekt aus, das die durch n n angegebenen Elemente des eingegebenen Objektes enthält. Die beiden Zahlen geben an, welche Elemente eingeschlossen werden sollen, vom ersten bis zum einschließlich letzten Element.

```
?piece 4 7 "Westfalen
tfal
?piece 2 4 [Nana Hans Michael Werner Andreas]
[Hans Michael Werner]
?make "sichere.pos piece 1 2 tf
?:sichere.pos
[50 50]
```

**plist** Name

Gibt die Merkmalliste des eingegebenen Namens aus. Eine Merkmalliste ist aus Merkmalpaaren zusammengesetzt. Das erste Element des Paares ist der Merkmalname, das zweite der Merkmalwert. Definierte Merkmale besitzen das Merkmal .DEF, definierte Variablen das Merkmal .APV. In der Erklärung von remprop finden Sie in der Prozedur merkm.weg ein Anwendungsbeispiel von plist in einer Prozedur.

```
?plist "dreieck
[.DEF [[grösse] [repeat 3 [fd :grösse rt 120]]]]
?plist "Stern
[.APV Abend]

?make "Vogel "blau
?:Vogel
blau
?plist "Vogel
[.APV BLAU]
?merkmal.weg "Vogel
?:Vogel
Vogel has no value
```

**po** Name | Namensliste

Gibt die Definitionen der angegebenen Prozeduren oder Variablen im Textfenster aus.

```
?po "quadrat
to quadrat :seite
repeat 4 [fd :seite rt 90]
end
?
```

**poall**

Gibt die Definitionen aller Prozeduren und Variablen im Arbeitsspeicher aus.

```
?poall
```

**pons**

Gibt die Namen und Werte aller Variablen im Arbeitsspeicher aus.

```
?pons
mittel is 40
lc is 20
uc is 80
Vogel is blau
Schlange is grün
```

**pops**

Listet die Namen und Definitionen aller Prozeduren im Arbeitsspeicher auf.

```
?pops
to zirkus
repeat 12 [flagge lt 30]
end
to quadrat :grösse
repeat 4 [fd :grösse rt 90]
```

```
end
to dreieck
repeat 3 [rt 120 fd 25]
end
```

## pots

Gibt die Titelzeile aller Prozeduren im Arbeitsspeicher oder den angegebenen Paketen aus.

```
?pots
to poly :seite :Winkel
to zählen :obj
to spi :seite :Winkel :Stufe
```

## pprop Name Merkmalname Merkmalwert

Trägt das angegebene Merkmalpaar in die Merkmalliste des Namens ein. Um Systemmerkmale und Ihre eigenen spezifischen Merkmale in die Merkmalliste eines Namens einzutragen, ist `pprop` der geeignete Befehl. Die Löschkommandos (`er`, `ern` und `erall`) löschen nur Dr. Logos Standard-Systemmerkmale aus den Merkmallisten. Deshalb müssen Sie `remprop` benutzen, um Ihre speziellen Merkmale aus einer Merkmalliste zu löschen, bevor Sie den Namen ganz aus dem Arbeitsspeicher löschen können. In Abschnitt 6.1 finden Sie weitere Information über Merkmale. Wenn Sie die Merkmalliste eines Namens ausgeben wollen, verwenden Sie `plist`.

```
?pprop "Henkersknecht ".APV "Scott
?Henkersknecht
Scott
?plist "Henkersknecht
[.APV Scott]
?plist "Scott
[]
?pprop "Ursula "Telefon [072-12345]
?plist "Ursula
[Telefon [072 - 12345]]
?
```

## pps

Der `pps`-Befehl zeigt die von Ihnen definierten Merkmalpaare aller Objekte an, die sich im Arbeitsspeicher befinden. Wenn Sie keine Merkmale zugeordnet haben, zeigt `pps` nichts an, und gibt nur das Prompt-Zeichen aus.

```
?pprop "Marlis "Grösse 1.72
?pps
Marlis's Grösse is 1.72
?plist "Marlis
[Grösse 1.72]
```

## pr Objekt (...)

Gibt die eingegebenen Objekte ohne die äußeren Klammern, aber mit einem Wagenrücklauf nach der letzten Eingabe auf dem Textfenster aus. Sie können eine beliebige Anzahl von Objekten eingeben, wenn Sie dem `pr`-Befehl eine geöffnete Klammer (vorstellen). Dabei wird nur nach dem letzten Objekt ein Wagenrücklauf (Return) ausgegeben. Vergleichen Sie mit `show` und `type`.

```
?pr [Das ist eine Meldung]
Das ist eine Meldung
?make "variable "dumme
?(pr [Das ist eine] :variable "Meldung
Das ist eine dumme Meldung
```

## \* n n (...)

**Produkt.** Der Stern `*` gibt das Produkt der eingegebenen Zahlen aus. Mehrere Zahlen können in Klammern eingeschlossen werden.

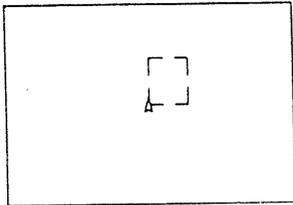
```
?* 7 6
42
?7 * 6
42
?* 4
Not enough inputs to *
```

```
?to kubik :#
>op (* :# :# :#)
>end
kubik defined
?kubik 5
27
?
```

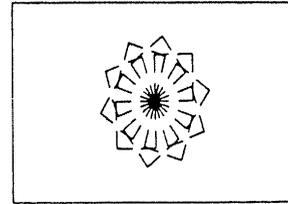
## pu

Hebt den Tintenstift des Igel ab, der Igel hört mit dem Zeichnen auf. Benutzen Sie pd, um wieder normal zu zeichnen.

```
?to drehe :grösse
>repeat 10 [rt 36 stift.hoch :grösse]
>end
drehe defined
?to stift.hoch :grösse
>repeat 4
> [fd :grösse pu
>   fd (:grösse / 1.618) pd
>   fd :grösse rt 90]
>end
stift.hoch defined
```



```
?stift.hoch 20
```



```
?drehe 20
```

## px

Der Igel kehrt in seiner logischen Spur die Farbe jedes vorher gefärbten Punktes zur Umkehr- bzw. Komplementärfarbe um. Der Igel hinterlässt eine Spur in der aktuellen Stiftfarbe, bis er auf einen bereits gefärbten Punkt trifft. Die Farbe des Punktes wird zu einer Kombination der Stiftfarbe und der existierenden Farbe umgewandelt. Wird ein und dieselbe Linie zweimal hintereinander im px-Modus gezogen, so werden die ursprünglichen Farben wiederhergestellt, die Linie verschwindet.

```
?show item 4 tf
?PD
?px show item 4 tf
?PX
?fd 20 pu fd 20
?pd setpc 3 fd 20
?px
?bk 80
?fd 80
?pd bk 1000
?
```

**Kanalstatus** gibt den Kanal an, der zur Tonerzeugung genutzt wird. Von den drei verfügbaren Kanälen wird normalerweise der erste eingesetzt. Mit `release` wird der belegte Kanal wieder freigegeben.

**Tonperiode** gibt die Höhe des Tones an. Je größer der eingegebene Werte ist, desto tiefer ist der Ton. Der nutzbare Bereich erstreckt sich von 5 bis 4095 - das mittlere `c` hat den Wert 239 und für den Kammerton `a` müssen Sie 142 eingeben. (Vergleichen Sie dazu auch die Notentabelle im Anhang G.)

**Tondauer** Der dritte Parameter gibt die zeitliche Länge des Tones an, wobei 1 Sekunde dem Wert 100 entspricht. Um einen Kammerton `a` mit einer Länge von 0,5 Sekunden zu *spielen*, müssen Sie `sound [1 142 50]` eingeben.

**Lautstärke** Dieser Parameter verändert die relative Lautstärke im Bereich von 0 bis 15, wobei der größte Wert auch dem lautesten Ton entspricht.

**ltst\_Hüllk** Die Hüllkurve der Lautstärke wird mit dem `env`-Befehl eingestellt. Sie können damit bis zu 16 (0 bis 15) *Lautstärkeschwankungen* definieren, die über die entsprechenden Nummern im `sound`-Befehl abgerufen werden.

**Ton\_Hüllk** Die Ton-Hüllkurve wird mit dem `ent`-Befehl eingestellt. Sie können damit bis zu 16 (0 bis 15) *Tonhöenschwankungen* definieren, die über die entsprechenden Nummern im `sound`-Befehl abgerufen werden.

**Rauschen** Dem Parameter Rauschen können Sie Werte von 1 bis 31 zuordnen, um unterschiedliche Rauscheffekte zu erzeugen.

```
?sound [1 478 50]
?sound [1 1911 100 10 1 1 15]
```

```
?to zweiklang
>sound [1 478 50]
>sound [2 213 50]
>end
zweiklang defined
?zweiklang
```

```
?to dreiklang
>sound [1 478 100]
```

```
>sound [2 213 100]
>sound [4 379 100]
>end
dreiklang defined
?dreiklang
```

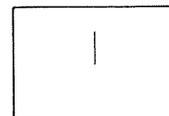
## SS

Öffnet ein Fenster für Textausgabe auf der Zeichenfläche. Schaltet den Bildschirmmodus von `fs` oder `ts` auf `ss` um. Der Kernbefehl `ss` setzt die Anzahl der Textzeilen im Fenster beim CPC6128 auf 5 und auf 8 beim Joyce. Diese Zahl können Sie mit `setsplit` verkleinern oder vergrößern. Beim Starten von Dr. Logo ist der `ss`-Modus eingeschaltet. In den Erklärungen von `fs` und `ts` finden Sie weitere Information.

```
?ts
?fs
?ss
```

## st

Macht den Igel sichtbar, wenn er versteckt ist. Wenn der Igel unsichtbar ist, zeichnet er schneller aber hebt sich nicht sichtbar von der Zeichnung ab. Um den Igel unsichtbar zu machen, geben Sie `ht` ein.



```
?ht
?fd 50
```



?st

## stop

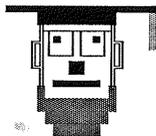
Unterbricht die Ausführung der gerade laufenden Prozedur und kehrt zur Hauptebene oder zur aufrufenden Prozedur zurück. Vergleichen Sie stop mit catch, throw, op und end.

```
?to verschwinde :objekt
>if empty? :objekt [stop]
>pr :objekt
>verschwinde bf :objekt
>end
verschwinde defined
?verschwinde HERRSCHER
HERRSCHER
ERRSCHER
RRSCHER
RSCHER
SCHER
CHER
HER
ER
R
?
```

## + nn (...)

**Summe.** Gibt die Summe der eingegebenen Zahlen aus. In der Erklärung von show finden Sie in addiere ein Anwendungsbeispiel von Summe bzw. + in einer Prozedur.

```
?+ 5 9
14
?+ 2.565 7.9
10.465
?(+ 6 4 -7 9 3 -2 8)
21
```



## T

### text Prozname

Gibt die aus zwei Elementen bestehende Definitionsliste der angegebenen Prozedur aus. Das erste Element enthält den (die) Eingabennamen der die Prozedur, sofern vorhanden. Wenn die Prozedur keine Eingabennamen hat, ist das erste Element eine leere Liste. Das zweite Element ist das .DEF-Merkmal der Merkmalliste des Prozedurnamens. Die Definitionsliste ist eine Textzeile, die kein end enthält. Der text-Befehl hat das gleiche Ausgabeformat, wie die Eingabe von define.

```
?define "quadrat [[seite] [repeat 4 [fd :seite rt 90]]
?po "quadrat
to quadrat :seite
repeat 4 [fd :seite rt 90]
end
?text "quadrat
[[seite] [repeat 4 [fd :seite rt 90]]]
?plist "quadrat
[.DEF [[seite] [repeat 4 [fd :seite rt 90]]]]
?
```

### tf

Gibt eine Liste aus, die die aktuellen Attribute des Igels beschreibt. Die 6 Elemente der Liste werden in der Form

```
[XORT YORT KURS STIFTZUSTAND STIFFFARBE IGELDA]
```

ausgegeben.

1. XORT steht für die X-Koordinate des Igels.
2. YORT steht für die Y-Koordinate des Igels.
3. KURS steht für die Kompassrichtung des Igels.

4. STIFTZUSTAND steht für pd, pe, pu, oder px.
5. STIFFFARBE steht für die Nummer der Stifffarbe des Igels.
6. IGELDA ist TRUE, wenn der Igel sichtbar ist, sonst FALSE.

```
?setpos [15 30] rt 60
?setpc 3 pe ht
?tf
?[15 30 RA 3 FALSE]
```

### thing Varname

Gibt den Inhalt bzw. den Wert der eingegebenen Variablen aus. thing ist mit dem Doppelpunkt ":" identisch, es gibt den Wert des Systemmerkmals .APV aus der Merkmalliste der Variablen aus. Allerdings verlangt poBEZUG, daß einem Variablennamen ein Doppelpunkt vorangeht.

```
?make "Geschmack "Schokolade
?thing "Geschmack
Schokolade
?:Geschmack
Schokolade
?:Schokolade
Schokolade has no value
?make "Schokolade "Halbbitter
?thing "Schokolade
Halbbitter
?:Geschmack
Schokolade
?thing :Geschmack
Halbbitter
?
```

### throw Name ✓

Führt die Zeile aus, die auf einen catch-Ausdruck mit gleichem Eingabennamen folgt. Sie können einen throw-Ausdruck nur zusammen mit einem catch-Ausdruck verwenden. Um einen catch-Ausdruck mit einem throw-Ausdruck zu verbinden, müssen catch und throw den gleichen Eingabennamen besitzen. Wenn throw ausgeführt wird, kehrt Dr. Logo zu der Prozedur zurück, die das catch-Kommando enthält, das den gleichen Namen wie das throw-Kommando besitzt. Dr. Logo führt dann

die dem catch-Ausdruck folgende Zeile aus. throw akzeptiert das spezielle Wort "TOPLEVEL, um von jedem Verschachtelungsgrad aus zum Prompt ? zurückzukehren (Vergleiche mit stop, end und op).

```
?to demo.catch
>make "Erfolg "FALSE
>catch "mein.fehler
> [mache.es :bis :Anweisungen]
>if :Erfolg = "TRUE [stop]
>make "Fehler1 error
>pr [Ich habe einen Fehler gefunden:]
>pr first bf :Fehler1
>end
demo.catch defined
?
```

```
?to mache.es :bis :Anweisungen
>;Anweisungen bis Aussage = TRUE
>label schleife
> if run :bis
>   [run :Anweisungen]
>   [throw "mein.fehler]
> make "Erfolg "TRUE
>go "schleife
>end
mache.es defined
?make "x 1
?make "bis [:x < 200]
?make "Anweisungen rl
fd 10 rt :x make "x :x + 3
?demo.catch
?make "x "r
?demo.catch
Ich habe einen Fehler gefunden:
bf deosn't like [] as input in demo.catch: pr first bf :Fehler1
?
```

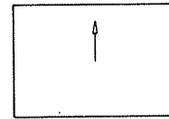
## to Prozname <Eingaben>

Zeigt Dr. Logo an, daß Sie mit der Definition einer Prozedur beginnen und schaltet den Prozedur-Editor ein (angezeigt durch das Prompt >). In Abschnitt 4.2 finden Sie weitere Information über die Benutzung des Prozedur-Editors.

```
?to fünfeck
>repeat 5 [fd 25 lt 72]
>end
fünfeck defined
?
```

## towards Koord\_Liste

Gibt die Blickrichtung aus, in der der durch die eingegebene Koordinatenliste bestimmte Punkt vom Standpunkt des Igels gesehen liegt. Um den Igel in die Richtung der eingegebenen Punktkoordinaten zu drehen, benutzen Sie towards in einem seth-Ausdruck. Wenn die eingegebene Koordinatenliste mit der aktuellen Position des Igels identisch ist, gibt towards 0 aus.



```
?cs
?fd 40
```



```
?seth towards [20 10]
```

**trace** ✓

Schaltet die Überwachung der Prozedurausführung und der Variablenzuweisung ein. Dabei ermöglicht `trace` die Überwachung von Prozeduren ohne Unterbrechung. Der Kernbefehl `trace` gibt den Namen jeder Prozedur beim ihrem Aufruf, den Namen und den Wert jeder Variable bei ihrer Definition und die Verschachtelungsstufe der Prozeduren während ihrer Ausführung aus. Um mit dem Interpretierer in Verbindung zu treten, benutzen Sie `Control-Z` bzw. `Alt-Z`, was die Ausführung der Prozedur unterbricht. Vergleichen Sie die Überwachung durch `trace` mit der Überwachung durch `watch`. Beide Überwachungsarten können gleichzeitig benutzt werden.

```
?to durchschnitt :zahlen
>make "anzahl count :zahlen
>make "summe addiere :zahlen
>op :summe / :anzahl
>end
durchschnitt defined
?to addiere :zahlen
>if empty? :zahlen [op 0]
>op (addiere bf :zahlen) +
> (first :zahlen)
>end
addiere defined
?durchschnitt [2 4 6]
4
?trace
?durchschnitt [2 4 6]
[1] Evaluating durchschnitt
[1] zahlen is [2 4 6]
[1] making "anzahl 3
[2] Evaluating addiere
[2] zahlen is [2 4 6]
[3] Evaluating addiere
[3] zahlen is [4 6]
[4] Evaluating addiere
[4] zahlen is [6]
[5] Evaluating addiere
[5] zahlen is []
[5] addiere returns op 0
[4] addiere returns op 6
[3] addiere returns op 10
[2] addiere returns op 12
[1] making "summe 12
[1] durchschnitt returns op 4
```

```
4
?notrace
?durchschnitt [4 6 8]
6
?
```

**ts**

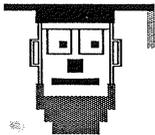
Macht den ganzen Bildschirm zum Textfenster. Schaltet den Bildschirmmodus von `ss` oder `fs` auf `ts` um. Wenn Sie sich im `ts`-Bildschirmmodus befinden und ein Graphikkommando ausführen, schaltet Dr. Logo automatisch auf `ss`-Modus um und löscht allen Text vom Bildschirm.

```
?fs
?ss
? to drehe.quadrat
>repeat 12 [repeat 4
> [fd 50 rt 90]
> rt 30]
>end
drehe.quadrat defined
?drehe.quadrat
?ts
```

**type** Objekt (...)

Gibt die eingegebenen Objekte im Textfenster aus, wobei die äußeren Klammern der Liste entfernt werden. `type` fügt zur letzten Eingabe keinen Wagenrücklauf hinzu. Vergleichen Sie `type` mit `pr` und `show`.

```
?type [Das ist die Position des Igels: ] show piece 1 2 tf
Das ist die Position des Igels: [-19 -21]
```



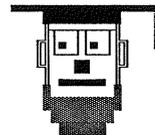
## U

**uc** Wort

Gibt das eingegebene Wort vollständig in Großbuchstaben aus. Akzente gehen bei der Umwandlung in Großbuchstaben verloren.

```
?uc "johannes
JOHANNES
?uc "AnnaMaria
ANNAMARIA
```

```
?to quiz
>type [Kannst du Okarina spielen?\ ]
>make "antwort rq
>if :antwort = [] [stop]
>if "J = uc first :antwort
> [pr [Toll, das ist Klasse!]]
> [pr [Ich auch nicht!]]
>quiz
>end
quiz defined
?quiz
Kannst du Okarina spielen? nicht allzu gut
Ich auch nicht!
Kannst du Okarina spielen? ja
Toll, das ist Klasse!
```



## W

**wait\***

Verzögert die Ausführung der Prozedur für um n/60 Sekunden.

```
?wait 100 pr "hallo
hallo

?to uhr
>repeat 10
> [(type "tick "\ ) sound [1 50 10]]
> wait 30
> pr "tack sound [1 70 8]
> wait 30]
>pr "BUMM! sound [1 2000 500]
>end
uhr defined
?uhr
tick tack
BUMM!
?
```

\*Der Kernbefehl wait ist nicht auf dem Joyce implementiert.

**watch** <Prozname | Proznamensliste> ✓

Schaltet die Ausdruck-für-Ausdruck Prozedurüberwachung ein. Der Kernbefehl `watch` legt vor der Ausführung jedes Ausdrucks eine Pause ein, um Ihnen den Kontakt mit dem Interpretierer oder Editor zu ermöglichen. Sie müssen die Return-Taste drücken, um den angezeigten Ausdruck auszuführen. Während der Pause können Sie mit dem Interpretierer von Dr. Logo in Kontakt treten, als wenn Sie beim Prompt `?` wären. Wenn Sie die Werte der Variablen auch automatisch ausgedruckt haben wollen, müssen Sie `trace` und `watch` zusammen benutzen. Die ausgegebene Zahl in eckigen Klammern gibt die Verschachtelungstiefe der gerade laufenden Prozedur an.

Der Kernbefehl `watch` überwacht alle Prozedurausführungen, bis Sie `nowatch` eingeben. Wenn Sie `watch` mit Prozedurnamen eingeben, überwacht er nur die angegebene(n) Prozedur(en), bis Sie `nowatch` eingeben. Beachten Sie, daß die in den unteren Beispielen benutzten Prozeduren bei der Erklärung von `trace` definiert wurden. Vergleichen Sie die Überwachung durch `trace` mit der Überwachung durch `watch`.

```
?durchschnitt [2 4 6]
```

```
4
```

```
?watch
```

```
?durchschnitt [2 4 6]
```

```
[1] In durchschnitt, make "anzahl count :zahlen
[1] In durchschnitt, make "summe addiere :zahlen
[2] In addiere, if empty :zahlen [op 0]
[2] In addiere, op ( addiere bf :zahlen ) + (first :zahlen )
[3] In addiere, if empty :zahlen [op 0]
[3] In addiere, op ( addiere bf :zahlen ) + (first :zahlen )
[4] In addiere, if empty :zahlen [op 0]
[4] In addiere, op ( addiere bf :zahlen ) + (first :zahlen )
[5] In addiere, if empty :zahlen [op 0]
[1] In durchschnitt, op :summe / :anzahl      4
```

```
?nowatch
```

```
?durchschnitt [4 6 8]
```

```
6
```

**where**

Gibt die Elementennummer des letzten erfolgreichen `memberp`-Ausdruckes aus. Sehen Sie auch in der Erklärung von `memberp` nach.

```
?memberp "u "Fluss
```

```
TRUE
```

```
?where
```

```
3
```

**window**

Ermöglicht es dem Igel, nach einem `wrap`- oder `fence`-Ausdruck außerhalb der Zeichenfläche zu zeichnen. Beim Starten von Dr. Logo ist der Bildschirmmodus `window` eingeschaltet. Sehen Sie auch die Erklärung von `wrap` und `fence`.

```
?fence
```

```
?to quadrat.spirale :seite
```

```
>if :seite > 200 [stop]
```

```
>repeat 4 [fd :seite rt 90]
```

```
>rt 20
```

```
>quadrat.spirale :seite + 5
```

```
>end
```

```
quadrat.spirale defined
```

```
?quadrat.spirale 20
```

```
Turtle out of bounds in quadrat.spirale:!
```

```
repeat 4 [fd :seite rt 90]
```

```
?window cs
```

```
?quadrat.spirale 20
```

**word** Wort Wort (...)

Gibt ein Wort aus, das sich aus den eingegebenen Worten zusammensetzt. (Beim Joyce müssen Sie statt \ das `ö` eingeben.)

```
?word "Hokus ":Pokus
```

```
HokusPokus
```

```
?40 "Tage
```

```
40Tage
```

```
?(word "ab "ra "ka "da "bra)
```

```
abrakadabra
```

```
?(word "Das "\ ist "\ ein "\ Wort)
Das ist ein Wort
```

## wordp Objekt

Gibt den Wert TRUE aus, wenn das eingegebene Objekt ein Wort oder eine Zahl ist, ansonsten wird FALSE ausgegeben.

```
?wordp "Narina
TRUE
?wordp 50
TRUE
?wordp [Wort]
FALSE
?wordp bf [grün rot gelb]
FALSE
?wordp first [blau rot gelb]
TRUE
```

## wrap

Schaltet den Bildschirmmodus von fence oder window auf wrap um. Der wrap-Befehl läßt den Igel auf der gegenüberliegenden Seite der Zeichenfläche erscheinen, wenn er die Grenze überschreitet. Um dem Igel das Zeichnen außerhalb der begrenzten Zeichenfläche zu ermöglichen, geben Sie window ein.

```
?wrap
?fd 180
?cs
?to tuch
>cs wrap
>setpc 3 rt 40 fd 10965
>setpc 2 rt 90 fd 5000
>setpc 1 pu rt 90 fd 6
>pd ht repeat 625
> [fd 3 lt 90 fd 1 rt 90
>   bk 3 lt 90 fd 1 rt 90]
>end
tuch defined
?tuch
```

## 9 Spezialwörterbuch

### Abschneiden

Verkürzen (normalerweise einer Zahl) durch Entfernen eines Teils.

### Abspeichern

Übertragung von Information aus dem RAM (Direktzugriffsspeicher) zur permanenten Speicherung auf eine Diskette.

### Arbeitsspeicherbereich

Dr. Logo löschrbarer Speicherbereich für Information, wie Prozeduren, Variablen oder Merkmallisten.

### ASCII

Abkürzung von American Standard Code for Information Interchange (Amerikanischer Standardcode für Informationsaustausch). ASCII ist ein Code zur computergerechten Darstellung der Zahlen, Buchstaben und Symbolen, die auf den meisten Tastaturen vorkommen.

### Ausdruck

Prozedurname plus Eingabeparameter (sofern welche verlangt werden).

### Ausführung einer Prozedur

Ein Prozess, in dem Dr. Logo die Ausdrücke einer Prozedur auswertet, bzw. ausführt.

### Ausgabe

Ein Objekt, das von einer Prozedur zurückgegeben wird. Oder: Daten, die der Prozessor zum Terminal, zum Drucker oder zur Diskette lenkt, nachdem die Bearbeitung beendet wurde. Oder: der Prozess des Sendens von Informationen zum Terminal, zum Drucker oder zur Diskette. Der Computer gibt das erwünschte Objekt oder die erwünschten Daten aus.

### Aussage

Ausdruck, der einen logischen Operator enthält und entweder TRUE (wahr) oder FALSE (falsch) ausgibt. Eine Aussage kann auch Eingabe für eine Dr.-Logo-Prozedur sein.

### Auswerten

Der Prozess, mit dem Dr. Logo einen Ausdruck ausführt.

**Begrenzer**

Leerzeichen und die unten aufgeführten Sonderzeichen, die verwendet werden, um Zeichen, Worte oder Zahlen vom übrigen Text abzuheben: "[", "]", "(", ")", ";", "<", ">", "+", "-", "\*", "/" und "^".

**Bereinigung**

siehe Säuberung.

**Bildpunkt**

Das kleinste, darstellbare Element auf dem Bildschirm; ein Punkt innerhalb der Zeichenzelle.

**Bildschirm**

Ausgabegerät, das den Inhalt des Graphik- oder Textpuffers anzeigt.

**Byte**

Speichereinheit im Hauptspeicher oder auf der Diskette, die aus 8 Bits besteht. Im ASCII-Code wird jedes Zeichen durch ein Byte dargestellt.

**Cursor**

Ein rechteckiges helles Zeichen auf dem Bildschirm. Der Cursor markiert die Position, die durch den nächsten Tastendruck angesprochen wird. Während Sie Zeichen eingeben, rückt der Cursor automatisch nach rechts. Sie können die Cursorposition auch mit der Tab-Taste und der Clr- bzw. Del-Taste verändern. Auf der graphischen Zeichenfläche wird der Aktionspunkt durch den Igel markiert.

**Datei**

Eine zusammenhängenden Folge von Information, die auf einem Datenträger abgespeichert ist.

**Dateiname**

Name, den Sie einer Datei zuweisen. Ein vollständiger Dateiname wird Dateibezeichnung genannt. Sie besteht aus drei Komponenten und zwar aus dem Laufwerknamen (ein Buchstabe und ein Doppelpunkt), dem Dateinamen (bis zu 8 Zeichen), und dem Dateityp (1 bis 3 Zeichen nach einem Punkt).

**Daten**

Information, die vom Computer erzeugt, geändert oder gespeichert wird.

**Dauerhafter Speicher**

Ein Platz außerhalb des Hauptspeichers, normalerweise eine Diskette, wo Daten gespeichert werden können. Wenn Sie eine Datei abspeichern, dann kopieren Sie sie aus dem Arbeitsspeicher auf eine Diskette.

**Diskette mit Nur-lese-Zugriff**

Diskette mit dem Attribut *Nur-lese-Zugriff*, das nur Lesen von der Diskette, aber kein Schreiben auf die Diskette erlaubt. Um einer Diskette das Attribut *Nur-lese-Zugriff* zuzuweisen, drücken Sie am oberen Teil der Diskette den Schieber des *Nur-lese-Zugriff-Ausschnitts* hinein bzw. legen die Schalter nach innen um.

**Diskette mit Schreib-lese-Zugriff**

Attribut einer Diskette, das sowohl Lesen von der Diskette, als auch Schreiben auf die Diskette erlaubt. Um einer Diskette das Attribut *Schreib-lese-Zugriff* zuzuweisen, ziehen Sie am oberen Teil der Diskette den Schieber des *Schreib-lese-Zugriff-Ausschnitts* heraus bzw. legen die Schalter nach außen um.

**Diskettenhülle**

Fester Kunststoffbehälter, der die Magnetscheibe umhüllt.

**Doppelseitig**

Diskettenformat, bei dem die Information auf beiden Seiten der Diskette gespeichert wird. Oder: Laufwerk, das auf beiden Seiten einer doppelseitigen Diskette lesen und schreiben kann.

**Dr. Logo**

Die Logo-Version von Digital Research: eine Programmiersprache mit Erweiterungen, die sie für kommerzielle oder Wissenschaftliche Zwecke anwendbar machen.

**Ebenennummer**

Nummer zwischen zwei eckigen Klammern, die am Anfang einer von watch oder trace ausgegebenen Zeile steht. Diese Nummer zeigt die Position der Prozedur auf dem Stapel an. Siehe Stapel.

**Editieren**

Prozess des Hinzufügens, Modifizierens oder Löschens von Text in einer vom Benutzer definierten Prozedur.

**Editor**

Dienstprogramm, das es Ihnen ermöglicht, Information im Textpuffer hinzuzufügen, zu modifizieren oder zu löschen. Siehe auch Zeileneditor, Prozedureditor und Texteditor.

**Eingabe**

Objekt, das eine Prozedur erwartet, um ihren Auftrag zu erfüllen. Die Eingabe folgt dem Prozedurnamen in der Anweisungsliste.

**Eingebettete Operation**

Operationen, in denen der Kernbefehlsname zwischen seinen Eingabeparametern steht, wie in "a + b", wo das Pluszeichen, "+" der eingebettete Operator ist.

**Einseitig**

Diskettenformat, in dem die Information auf nur einer Seite der Diskette gespeichert wird. Oder: ein Laufwerk, das nur von einer Seite der Diskette lesen und nur auf eine Seite der Diskette schreiben kann.

**Element**

Ein Teil eines Objektes. Ein Ziffer ist zum Beispiel Element einer Zahl, oder ein Zeichen ist Element eines Wortes, und ein Wort ist Element einer Liste.

**end**

Schlüsselwort, das die Beendigung der Definition einer Prozedur anzeigt. Das Wort end muß allein am Anfang der letzten Zeile der Prozedur stehen.

**Ersatzzeichen**

Das Fragezeichen "?". Mit diesem Zeichen wird von Dr. Logo ein Muster zusammengestellt, die bei der Suche nach übereinstimmenden Dateinamen in einem Inhaltsverzeichnis verwendet wird. Ein Fragezeichen in einem Dateinamen erzeugt einen mehrdeutigen Dateinamen. Siehe unter mehrdeutiger Dateiname.

**Fehlermeldung**

Eine Meldung, die Ihr Terminal ausgibt, wenn Sie ein Kommando eingegeben haben, das der Computer nicht versteht, oder wenn der Computer Fehler hat.

**Fehlersuche und Fehlerbeseitigung**

Vorgang der Suche und Beseitigung von Fehlern in einer Prozedur

**Fenster**

Sichtbarer Teil des Textpuffers. Siehe auch Textfenster.

**Flexible Diskette**

Biegbare magnetische Scheibe zur Informationsspeicherung. Eine flexible Diskette ist eine dünne Scheibe aus Mylarfolie, die mit magnetischen Teilchen beschichtet ist. Das Laufwerk bringt die Diskette innerhalb der Kunststoffhülle in Rotation.

**Funktion**

Operation oder Auftrag, der ausgeführt wird, wenn der Prozedurname aufgerufen wird. Das Ergebnis einer Funktion ist immer ein einfacher Wert. Siehe auch Operation.

**Funktionstasten**

Mehrzweckstasten, die Sie so programmieren können, daß ganze Kommandozeilen aus gespeichert werden können.

**Ganze Zahl**

Positive oder negative Zahl ohne Dezimalpunkt.

**Geteilter Bildschirm**

Anzeigemodus, in dem ein Textfenster auf der Zeichenfläche eröffnet wird.

**Graphischer Puffer**

Platz im Hauptspeicher, in dem graphische Daten gespeichert werden. Sie können den graphischen Puffers auf einem Bildschirm ausgeben. Dr. Logo ermöglicht Ihnen, mit Hilfe des Igels Zeichnungen zu erstellen.

**Graphische Anzeige**

Sichtbare Darstellung, die mathematische Kurven, Diagramme, Graphiken oder Landkarten illustriert und abbildet. Dr. Logo ermöglicht Ihnen, mit dem Igel graphische Zeichnungen zu erstellen.

**Graphische Zeichenfläche**

siehe Zeichenfläche.

**Grundwort**

siehe Kernbefehl.

**Hauptebene**

Zustand des Interpretierers, wenn Dr. Logo ein Fragezeichen ausgibt, sich keine Prozeduren im Stapel befinden, und die Ebenennummer ist 0.

**Hauptprozedur**

Eine Prozedur, die nie von anderen Prozeduren aufgerufen wird. Sie wird auch Hauptebenen-Prozedur genannt. Eine Hauptprozedur können Sie durch Eingabe ihres Namens als Antwort auf Dr. Logos Prompt "?" starten.

**Igel**

Graphisches Symbol, das auf Dr. Logos Zeichenfläche wie ein Cursor funktioniert.

**Igelgraphik**

Dr. Logos graphische Umgebung. Die Programmiersprache Dr. Logo ermöglicht Ihnen, durch Steuerung der Igelbewegungen Zeichnungen auf dem Bildschirm zu erstellen. Während seiner Bewegung hinterläßt der Igel eine Spur auf der Zeichenfläche.

**Igelschritt**

Die Entfernung zwischen zwei benachbarten Bildpunkten.

**Inhaltsverzeichnis**

Liste des Disketteninhalts. Verwenden Sie den Kernbefehl *dir*, um die Liste aller Dr.-Logo-Dateien auf einer Diskette zu erhalten.

**Initialisieren**

Das Verfahren der Vorbereitung oder Formatierung einer Diskette, um auf sie Informationen schreiben und von ihr Informationen lesen zu können.

**Interpretierer**

Computerprogramm, das jede Zeile direkt auswertet, wie sie über die Tastatur eingegeben wird und das jede Zeile einer Prozedur dann auswertet, wenn die Prozedur ausgeführt wird. Dr. Logo ist eine interpretierende Programmiersprache.

**Kernbefehle**

Prozeduren, Operationen oder Kommandos, aus denen Dr. Logo besteht. Die eingebauten Prozeduren, auch Grundwörter oder Primitive genannt.

**Kommando**

Anweisung, die Dr. Logo eine Tätigkeit einleiten läßt.

**Kommentar**

Anmerkung oder Erklärung, die vom Rest der Definition der Prozedur durch einen Semikolon (;) abgetrennt ist. Der Kommentartext wird von Dr. Logo ignoriert.

**Konstante**

Objekt, Wort, Zahl, oder Liste, dessen Wert während der Ausführung einer Prozedur nicht geändert wird.

**Kontrollzeichen**

Eine nicht druckbare Zeichenkombination, die ein einfaches Kommando an Dr. Logo sendet. Um ein Kontrollzeichen einzugeben, halten Sie die Ctrl-Taste gedrückt und drücken Sie die entsprechende Zeichentaste.

**Laden**

Kopieren von Prozeduren oder anderer Information von einer Diskette, wo sie permanent gespeichert sind, in den Speicherpuffer oder Arbeitsspeicher des Computers.

**Laufwerk**

Peripheres Gerät, das Information auf Platten schreiben und von dort lesen kann. Siehe Platte.

**Laufwerk mit Nur-lese-Zugriff**

Laufwerk mit dem Attribut *Nur-lese-Zugriff*, das nur das Lesen einer Datei von der Diskette, nicht aber Hinzufügen neuer Dateien, Löschen, Verändern oder Umbenennen von Dateien auf der Diskette erlaubt.

**Lesen**

Prozess der Übertragung der vorher gespeicherten Information von einem Speichermedium, wie zum Beispiel einer Diskette in den Hauptspeicher des Computers.

**Lichtgriffel**

Eingabegerät, das es ermöglicht, ohne Tastatur mit dem Computer zu kommunizieren. Der Lichtgriffel schickt die Koordinaten innerhalb des Textfensters an den Computer, wenn Sie mit der Spitze des Lichtgriffels die Bildschirmoberfläche berühren.

**Liste**

Ein Objekttyp, der als Eingabe zu einer Dr.-Logo-Prozedur verwendet wird. Eine Folge von Objekten (Worten, Zahlen oder Listen), die voneinander durch Leerzeichen getrennt sind und zwischen eckigen Klammern ([ und ]) eingeschlossen sind.

**Logischer Operator**

Mathematisches Symbol, das in Aussagen verwendet wird. Ausdrücke, die bei Auswertung WAHR oder FALSCH ergeben. Es existieren die Operatoren *gleich* (=), *kleiner als* (<) und *größer als* (>), sowie die Kernbefehle *and*, *not* und *or*.

**Logo**

Name einer Programmiersprache, der aus dem griechischen Wort "logos" (= Wort) abgeleitet wurde. Logo wurde konzipiert, um Kindern zu ermöglichen, ihre Ideen auf Computern zu entfalten.

**Magnetplatte**

siehe Platte.

**Mehrdeutiger Dateiname**

Dateiname, der Ersatzzeichen enthält, um gleichzeitig mehr als eine Datei anzusprechen. In Dr. Logo wird das Fragezeichen, "?" als Ersatzzeichen verwendet. Wenn Sie ein Zeichen in einem Dateinamen durch ein Fragezeichen "?" ersetzen, stellen Sie damit Dr. Logo ein Muster zur Verfügung, womit er Übereinstimmungen suchen kann. Wenn Sie zum Beispiel DIR "P???????" eingeben, gibt Dr. Logo alle Dateinamen zurück, die aus 8 Zeichen bestehen und mit dem Buchstaben P beginnen. Siehe auch Ersatzzeichen.

**Merkmal**

Eigenschaft eines Objekts, einer Prozedur

**Merkmalsliste**

Liste von Merkmalpaaren, die Eigenschaften von Objekten oder Prozeduren repräsentieren. Wenn Sie Dr. Logos Systemmerkmale oder Ihre eigenen Merkmale in Merkmallisten eintragen wollen, verwenden Sie den Kernbefehl *pprop*. Wenn Sie eine Merkmalliste erhalten wollen, verwenden Sie den Kernbefehl *plist*.

**Merkmalname**

Das erste Element eines Merkmalpaares: die Marke, die die Eigenschaft des Objekts, der Prozedur repräsentiert.

**Merkmalpaar**

Die beiden Teile eines Merkmals. Das erste Element eines Merkmalpaares ist der Merkmalname, das zweite Element ist sein Wert. In der Merkmalliste einer Prozedur zum Beispiel gibt es ein Merkmalspaar, das aus dem Merkmalsnamen *.DEF* und aus dem Wert von *.DEF*, d.h. aus der eigentlichen Definition der Prozedur besteht.

**Merkmalwert**

Das zweite Element eines Merkmalpaares: die Eigenschaft des Objekts oder der Prozedur.

**Monitor**

siehe Bildschirm.

**Name**

Objekttyp, der als Eingabe für Dr.-Logo-Prozeduren verwendet wird. Oder: Schlüsselwort, das eine Prozedur, eine Variable, eine Datei oder eine Diskette identifiziert.

**Numerische Konstante**

Eine reelle oder ganze Zahl, die während der Ausführung der Prozedur nicht verändert wird.

**Nur-lese-Zugriff**

Attribut einer Diskette oder eines Laufwerkes, das nur das Lesen von der Diskette, nicht aber das Schreiben auf die Diskette erlaubt.

**Objekt**

Eingabeparameter für Dr.-Logo-Prozeduren. Ein Logo-Objekt kann ein Wort, eine Zahl, eine Liste oder etwas anderes sein, das kein Prozedurname ist. Variablen können Objekte repräsentieren.

**Operation**

Prozedur, die ein Objekt ausgibt, oder ein mathematischer Prozeß, wie Addieren, Subtrahieren, Multiplizieren oder Dividieren. In Dr. Logo ist eine Operation einer Funktion äquivalent.

**Operatorpriorität**

Reihenfolge, in der Dr. Logo arithmetische Operationen ausführt.

**Optionale Eingabe**

Eingabe, die nicht unbedingt verlangt wird. In der Beschreibung der Syntax der Kernbefehlausdrücke (Kapitel 8) sind optionale Eingaben in spitzen Klammern ("*<*" und "*>*") eingeschlossen.

**Platte**

Magnetisches Medium zur Informationsspeicherung. Wird oft Diskette genannt. Programme und andere Informationen werden ähnlich gespeichert und abgerufen, wie Musik von einem Tonband. Der Begriff Diskette steht für flexible Disketten mit einem Durchmesser von 3, 3 ½, 5 ¼ oder 8 Zoll. Der Begriff Laufwerk kann Diskette, eine Wechselplatte oder Festplatte bedeuten.

**Programm**

Ein kompletter Satz von Prozeduren, die einem Computer angeben, wie er einen bestimmten Auftrag zu erfüllen hat.

**Programmfehler**

Fehler in einer Prozedur, der verhindert, daß die Prozedur wie erwartet ausgeführt wird.

**Programmieren**

Eine Dr.-Logo-Prozedur definieren.

**Prompt**

Ein Signal, das auf dem Bildschirm ausgegeben wird, um dem Benutzer zu zeigen, daß Dr. Logo Eingaben erwartet. Ein Prompt kann ein Symbol sein, wie "?", ">" oder "!". Ein Prompt kann auch eine Meldung sein, wie *I don't know how to think*. Nach einem Prompt erwartet Dr. Logo eine Antwort vom Benutzer. Wenn Sie zum Beispiel das Prompt "!" sehen, haben Sie außerordentlich wenig freie Knoten im Arbeitsspeicher. Ihre Antwort muß die Eingabe des Kernbefehls *recycle* sein, um das Säuberungsprogramm aufzurufen. Siehe auch Prompt des Interpretierers und Prompt des Prozedureditors.

**Prompt des Interpretierers**

Ein Fragezeichen (?) zum Anzeigen des Wartezustandes; Dr. Logos Interpretierer wartet auf eine Eingabe.

**Prompt des Prozedureditors**

Das Größer-Zeichen (>) zum Anzeigen des Wartezustandes; Dr. Logo erwartet, daß Sie eine Zeile zur Definition einer Prozedur eingeben. Dr. Logo wertet die nach dem Prompt ">" eingegebenen Ausdrücke nicht sofort nach der Eingabe aus.

**Protokolliergerät**

Ein Gerät, das zum Ausdrucken der Daten benutzt wird. Ein Drucker ist ein Protokolliergerät.

**Prozedur**

Eine Folge von Ausdrücken, die Dr. Logo sagen, wie er einen Auftrag erfüllen soll.

**Prozedureditor**

Ein Dienstprogramm, das Ihnen ermöglicht, Dr. Logo einen neuen Auftrag beizubringen, d.h. eine neue Prozedur zu definieren. Der Prozedureditor meldet sich mit dem Größer-Zeichen (>), einem Prompt, das erscheint, wenn Sie *to* als Antwort auf das Prompt des Interpretierers, d.h. auf das Fragezeichen (?) eingeben. Im Prozedureditor sind alle Kontrollzeichenkommandos des Zeileneditors funktionsfähig.

**Puffer**

Bereich im Hauptspeicher, in dem Information während der Datenübertragung vorläufig gespeichert wird.

**Reelle Zahl**

Numerischer Wert, der mit einem Dezimalpunkt (Dezimalkomma) eingegeben und intern in Fließkommadarstellung gespeichert wird.

**Rekursive Prozedur**

Eine Prozedur, die durch sich selbst definiert ist.

**Return**

Informationsübergabe an den Computer. Der Computer kann die Information, die nach seinem letzten Prompt über die Tastatur eingetippt wurde, weder erkennen, noch verarbeiten, solange Sie die Return-Taste nicht gedrückt haben. Entspricht der Wagenrücklaufaste auf der Schreibmaschinentastatur.

**Säuberung**

Eine Operation, die Knoten in Ihrem Arbeitsspeicher freisetzt, damit sie benutzt werden können.

**Sicherungskopie**

Ein Duplikat von Daten, das verwendet werden kann, wenn das Original verlorengeht, zerstört oder ungewollt geändert wird.

**Sichtbares Feld**

Der Teil des Text- oder Graphikpuffers, der auf dem Bildschirm ausgegeben wird.

**Speicher**

Der Teil des Computersystems, der zur zeitweiligen Speicherung der Information dient. Wird auch RAM (Direktzugriffsspeicher) oder Arbeitsspeicher genannt. Dr. Logo wird automatisch in den Speicher geladen, wenn Sie Ihr System starten. Nach dem Laden von Dr. Logo teilt Ihnen dieser einen Teil des Speichers als Arbeitsspeicher zu.

**Stapel**

Bereich im Speicher, der vom Interpretierer zur Verwaltung des Ablaufs der verschiedenen Prozeduren verwendet wird.

**Standardwert**

Wert, der zugewiesen wird, wenn keine anderslautende Anweisung vorliegt. Beim Starten des Computers ist zum Beispiel das Laufwerk A: das Standardlaufwerk. Wenn Sie in den Eingaben kein Laufwerk angeben, sucht Dr. Logo die Dateien auf der Diskette im Standardlaufwerk.

**Steuerknüppel**

Eingabegerät, das ermöglicht, ohne Tastatur mit dem Computer zu kommunizieren. Steuerknüppel oder Joysticks werden gewöhnlich für Computerspiele verwendet. Die Koordinatenskalierung, die der Steuerknüppel erzeugt, hängt vom Gerätentyp an. Dr. Logo unterstützt bis zu zwei Steuerknüppel mit je zwei Knöpfen.

**Syntax**

Formatvorschrift bei Eingabe eines Ausdrucks.

**Teilbildschirm**

siehe geteilter Bildschirm.

**Temporärer Speicher**

Speicherbereiche, in denen die Daten nicht dauerhaft festgelegt sind.

**Texteditor**

Ein Dienstprogramm, das ermöglicht, eine Prozedur zu definieren oder eine Prozedur zu modifizieren, ohne die komplette Definition noch einmal in vollem Umfang eingeben zu müssen. Im Texteditor können Sie bestimmte Kontrollzeichen benutzen,

mit denen Sie den Cursor innerhalb der Prozedur von Zeile zur Zeile bewegen können. Auch die Kontrollzeichenkommandos des Zeileneditors stehen Ihnen zur Verfügung. Den Texteditor können Sie durch Eingabe der Kernbefehle *ed*, *edall*, oder *edf* als Antwort auf Dr. Logos Prompt "?" aufrufen.

### Textfenster

Sichtbarer Teil des Textpuffers. Der Teil des Bildschirms, auf dem Textdaten ausgegeben werden. Dr. Logos Kernbefehl *ts* (textscreen) öffnet ein Textfenster auf der Zeichenfläche.

### Textpuffer

Speicherbereich, in dem Textdaten gespeichert werden. Das sichtbare Fenster des Textpuffers können Sie auf dem Bildschirm ausgeben.

### trace

Ein Befehl, die zur Fehlersuche während der Ausführung einer Prozedur verwendet wird. *trace* gibt den Namen jeder Prozedur beim Aufruf und den Wert jeder Variablen bei der Definition aus. *trace* ermöglicht die Überwachung von Einzelheiten während der Ausführung einer Prozedur, ohne die Ausführung unterbrechen zu müssen. Die *trace*-Option können Sie mit dem Kommando *notrace* stoppen.

### Urladen

Vorgang des Startens des Computers.

### Variable

Name, dem Dr. Logo ein Objekt als Wert zuweisen kann. Eine Variable kann man sich wie einen Behälter vorstellen, der einen Wert enthalten kann.

### Vorangestellte Operation

Operation, in der der Kernbefehlsname oder der Operator den Eingaben vorangeht, wie in "- a b", wo das Minuszeichen der vorangestellte Operator ist.

### Vorgegeben

siehe Standardwert.

### watch

Befehl, der zur Fehlersuche während der Ausführung einer Prozedur verwendet wird. Die *watch*-Option gibt den Namen der Prozedur und den Ausdruck zeilenweise aus, den der Interpretierer als Nächsten auswerten wird. Bevor die betreffende Zeile ausgeführt wird, wartet Dr. Logo so lange, bis Sie die Return-Taste drücken. *watch* ermöglicht Ihnen, mit dem Interpretierer oder dem Editor während der Ausführung einer Prozedur zu kommunizieren.

### Wert

Eine Größe; zum Beispiel eine ganze oder reelle Zahl. Das Objekt, das einer Variablen oder einem Merkmalsnamen zugewiesen wurde.

### Wort

Ein Objekttyp, der als Eingabe für eine Dr.-Logo-Prozedur verwendet wird. Eine Gruppe von Null oder mehr aufeinanderfolgenden Zeichen, die von den anderen Zeichen in der Zeile durch Begrenzer getrennt sind.

### Zahl

Objekttyp, der als Eingabe für Dr.-Logo-Prozeduren verwendet wird. In Dr. Logo ist eine Zahl eine Art Wort und kann auch als Variablenname verwendet werden, wenn ihr, wie bei "8, ein Anführungszeichen vorangestellt wird. Eine Zahl kann nicht als erstes oder einziges Zeichen in einem Prozedurnamen stehen. In arithmetischen Operationen können Sie positive oder negative Dezimalzahlen mit bis zu 15 signifikanten Stellen verwenden.

### Zeichen

Ein einfacher Buchstabe, eine Zahl, ein Symbol, ein Leerzeichen oder Satzzeichen. Ein Zeichen ist normalerweise in einem Byte im Hauptspeicher oder auf der Diskette gespeichert.

### Zeichenfläche

Sichtbare Anzeige des graphischen Puffers. Der Teil des Bildschirms, auf dem Graphik ausgegeben wird.

### Zeichenzelle

Flächeneinheit auf dem Bildschirm, die ein Zeichen beinhalten kann. Jede Zeichenzelle befindet sich im Schnittpunkt einer horizontalen Zeile und einer vertikalen Spalte. Dr. Logo ermöglicht, die Hinter- und Vordergrundfarbe jeder Zelle der Zeichenfläche zu steuern.

### Zeileneditor

Ein Dienstprogramm, mit dessen Hilfe Sie die aktuelle Textzeile modifizieren können, ohne dazu die ganze Zeile löschen und erneut eingeben zu müssen. Dr. Logo ermöglicht die Verwendung einiger Kontrollzeichen oder Funktionstasten, um den Cursor über die Zeile nach links und rechts zu bewegen und dort Modifikationen vorzunehmen, bevor Sie die Return-Taste drücken.

### Zufallszahl

Eine aus einer Zahlenmenge zufällig ausgewählte Zahl. Der Kernbefehl *random* gibt eine solche Zahl zurück.

## A Kontrollzeichenkommandos

**Tabelle A-1a:** Kontrollzeichen- und Funktionstastekommandos beim CPC6128

Kommando	Operation
<b>Clr</b> (Control-D)	Löscht das Zeichen, das durch den Cursor markiert ist.
<b>Copy</b> (Control-C)	Beendet das Texteditieren, verläßt den Texteditor und aktualisiert den Arbeitsspeicherbereich von Dr. Logo mit allen Prozeduren und Variablen aus dem Texteditorpuffer. Wenn Kommandos im Editierpuffer abgespeichert sind, werden sie ausgeführt.
<b>Del</b> (Control-H)	Löscht das Zeichen links vom Cursor.
<b>Esc</b> (Control-G)	Außerhalb des Texteditors: Sofortige Beendigung der Ausführung der aktuellen Prozedur. Innerhalb des Texteditors: Verläßt den Texteditor, läßt aber den Arbeitsspeicherbereich von Dr. Logo unverändert. Ignoriert den ganzen Editierpuffer, einschließlich aller Änderungen, die Sie während der Editiersitzung vorgenommen haben. Sie können den verlassenen Editierpuffer durch Eingabe von ed wiederherstellen, wenn Sie nach Drücken der Esc-Taste noch kein anderes Zeichen eingegeben haben.
<b>Tab</b> (Control-I)	Fügt einen Tabulator ein. Der Cursor springt zum nächsten Tabulator, der in der betreffenden Zeile gesetzt wurde.
<b>→</b> (Control-F)	Der Cursor springt um ein Zeichen vorwärts, d.h. um eine Position

← (Control-B)	Der Cursor springt um ein Zeichen zurück; d.h. um eine Position nach links.
↓ (Control-N)	Der Cursor springt zur nächsten Zeile; er bewegt sich um eine Zeile nach unten in Richtung zum Ende des Puffers.
↑ (Control-P)	Der Cursor springt um eine Zeile zurück, d.h. nach oben, in der Richtung zum Anfang des Puffers
Control-→ (Control-E)	Der Cursor springt zum Ende der Zeile.
Control-← (Control-A)	Der Cursor springt zum Anfang der Zeile.
Control-↓ (Control-V)	Gibt die nächste Seite des Textes im Texteditorpuffer auf dem Bildschirm aus, d.h. die nächsten Zeilen in Richtung zum Ende des Puffers.
Control-↑ (Control-U)	Gibt die vorherige Seite des Textes im Texteditorpuffer auf dem Bildschirm aus, d.h. die vorherigen Zeilen in Richtung zum Anfang des Puffers.
Control-K	Löscht den Rest der Zeile, d.h. es löscht alle Zeichen rechts vom Cursor bis zu 133 Zeichen. Die gelöschten Zeichen werden im Zeileneditier-Puffer gespeichert und können mit Control-Y wiederhergestellt werden.

Control-L	Außerhalb des Texteditors: Ausgabe einer vollen Zeichenfläche. Der Bildschirm steht für die Igelgraphik allein zur Verfügung. Innerhalb des Texteditors: Stellt die Bildschirmanzeige so ein, daß die Zeile, in der sich der Cursor befindet, in der Mitte des Bildschirms erscheint. Wenn die Entfernung zwischen dem Pufferanfang und dem Cursor kleiner ist, als 12 Zeilen, wird das Textfenster beim Drücken von Control-L einfach erneut ausgegeben.
Control-M	Erzeugt einen Wagenrücklauf (Return); gibt die Information in den Rechner ein.
Control-0	Schafft eine neue Zeile. Eingabe von Control-0 entspricht der unmittelbar aufeinanderfolgenden Eingabe der Return-Taste und ↑.
Control-Q	Generiert ebenso wie die entsprechende Taste den umgekehrten Schrägstrich (\), der Dr. Logo einen Begrenzer als Textbestandteil betrachten läßt.
Control-R	Der Cursor springt zum Anfang des Texteditorpuffers.
Control-S	Teilt den Bildschirm; eröffnet ein Textfenster auf dem Graphikbildschirm.
Control-T	Schaltet in Textmodus um: der Bildschirm dient zur Textausgabe. Drücken Sie zweimal Control-T, damit der Cursor auch oben bleibt!
Control-W	Unterbricht das Rollen eines Textes; wartet bis zum Drücken der nächsten Taste, um die Ausgabe fortzusetzen.
Control-X	Der Cursor springt zum Ende des Texteditorpuffers.

Control-Y	Stellt einen Text aus dem Puffer wieder her, d.h. die letzte in den Puffer mit der Return-Taste oder mit Control-K eingespeicherte Zeile wird wieder angezeigt.
Control-Z	Unterbricht die momentan ausgeführte Prozedur und gibt das pause-Prompt aus, um interaktive Fehlersuche zu ermöglichen. Geben Sie oo ein, wenn Sie die Ausführung der Prozedur fortsetzen möchten; geben Sie throw "TOPLEVEL ein, wenn Sie zur höchsten Ebene umsteigen möchten oder geben Sie stop ein, wenn Sie zur vorherigen Ebene zurückkehren möchten.

Tabelle A-1b: Kontrollzeichen- und Funktionstastekommandos beim Joyce

Kommando	Operation
Copy, Einbl (Alt-Y)	Stellt einen Text aus dem Puffer wieder her, d.h. die letzte in den Puffer mit der Return-Taste oder mit Alt-K eingespeicherte Zeile wird wieder angezeigt.
[Del→] (Alt-D)	Löscht das Zeichen, das durch den Cursor markiert ist.
[+Del] (Alt-H)	Löscht das Zeichen links vom Cursor.
Exit (Alt-C)	Beendet das Texteditieren, verläßt den Texteditor und aktualisiert den Arbeitsspeicherbereich von Dr. Logo mit allen Prozeduren und Variablen aus dem Texteditorpuffer. Wenn Kommandos im Editierpuffer abgespeichert sind, werden sie ausgeführt.
[Seite] (Alt-V)	Gibt die nächste Seite des Textes im Texteditorpuffer auf dem Bildschirm aus, d.h. die nächsten Zeilen in Richtung zum Ende des Puffers.

Stop  
(Alt-G)

Außerhalb des Texteditors: Sofortige Beendigung der Ausführung der aktuellen Prozedur. Innerhalb des Texteditors: Verläßt den Texteditor, läßt aber den Arbeitsspeicherbereich von Dr. Logo unverändert. Ignoriert den ganzen Editierpuffer, einschließlich aller Änderungen, die Sie während der Editiersitzung vorgenommen haben. Sie können den verlassenen Editierpuffer durch Eingabe von ed wiederherstellen, wenn Sie nach Drücken der Stop-Taste noch kein anderes Zeichen eingegeben haben.

Tab  
(Alt-I)

Fügt einen Tabulator ein. Der Cursor springt zum nächsten Tabulator, der in der betreffenden Zeile gesetzt wurde.

[Zeile Ez]  
(Alt-E)

Der Cursor springt zum Ende der Zeile.

Shift-[Dok]  
(Alt-X)

Der Cursor springt zum Ende des Texteditorpuffers.

Shift-[Zeile Ez]  
(Alt-A)

Der Cursor springt zum Anfang der Zeile.

Alt-[Seite]  
(Alt-U)

Gibt die vorherige Seite des Textes im Texteditorpuffer auf dem Bildschirm aus, d.h. die vorherigen Zeilen in Richtung zum Anfang des Puffers.

Alt-Shift-[Dok]  
(Alt-R)

Der Cursor springt zum Anfang des Texteditorpuffers.

Alt-[Del→]  
(Alt-K)

Löscht den Rest der Zeile, d.h. es löscht alle Zeichen rechts vom Cursor bis zu 133 Zeichen. Die gelöschten Zeichen werden im Zeileneditier-Puffer gespeichert und können mit Alt-Y wiederhergestellt werden.

→ (Alt-F)	Der Cursor springt um ein Zeichen vorwärts, d.h. um eine Position
← (Alt-B)	Der Cursor springt um ein Zeichen zurück; d.h. um eine Position nach links.
↓ (Alt-N)	Der Cursor springt zur nächsten Zeile; er bewegt sich um eine Zeile nach unten in Richtung zum Ende des Puffers.
↑ (Alt-P)	Der Cursor springt um eine Zeile zurück, d.h. nach oben, in der Richtung zum Anfang des Puffers
Alt-L	Außerhalb des Texteditors: Ausgabe einer vollen Zeichenfläche. Der Bildschirm steht für die Igelgraphik allein zur Verfügung. Innerhalb des Texteditors: Stellt die Bildschirmanzeige so ein, daß die Zeile, in der sich der Cursor befindet, in der Mitte des Bildschirms erscheint. Wenn die Entfernung zwischen dem Pufferanfang und dem Cursor kleiner ist, als 12 Zeilen, wird das Textfenster beim Drücken von Alt-L einfach erneut ausgegeben.
Alt-M	Erzeugt einen Wagenrücklauf (Return); gibt die Information in den Rechner ein.
Alt-O	Schafft eine neue Zeile. Eingabe von Alt-O entspricht der unmittelbar aufeinanderfolgenden Eingabe der Return-Taste und ↑.
Alt-Q	Generiert ebenso wie die entsprechende Taste den Großbuchstaben Ö, der Dr. Logo einen Begrenzer als Textbestandteil betrachten läßt.
Alt-S	Teilt den Bildschirm; eröffnet ein Textfenster auf dem Graphikbildschirm.

Alt-T	Schaltet in Textmodus um: der Bildschirm dient zur Textausgabe. Drücken Sie zweimal Alt-T, damit der Cursor nach oben geht!
Alt-W	Unterbricht das Rollen eines Textes; wartet bis zum Drücken der nächsten Taste, um die Ausgabe fortzusetzen.
Alt-Z	Unterbricht die momentan ausgeführte Prozedur und gibt das pause-Prompt aus, um interaktive Fehlersuche zu ermöglichen. Geben Sie co ein, wenn Sie die Ausführung der Prozedur fortsetzen möchten; geben Sie throw "TOPLEVEL ein, wenn Sie zur höchsten Ebene umsteigen möchten oder geben Sie stop ein, wenn Sie zur vorherigen Ebene zurückkehren möchten.

## B Dr. Logos Systemkernbefehle

Dr. Logos Systemkernbefehle werden immer von einem Punkt eingeleitet. Diese Kernbefehle ermöglichen dem Benutzer, Hardwareperipherie zu steuern oder den Inhalt des Hauptspeichers zu untersuchen.

Wichtig: Ein Teil dieser Systemkernbefehle sollten nur von Programmierern verwendet werden, die über Erfahrungen in Assemblersprache verfügen.

Tabelle B-1: Dr. Logos Systemkernbefehle

Kernbefehl	Eingaben	Definition
<code>.contents</code>		Gibt den Inhalt von Dr. Logos Symbolspeicher aus.
<code>.deposit</code>	<code>n1 n2</code>	Trägt <code>n2</code> in die durch <code>n1</code> festgelegte absolute Hauptspeicherzelle im aktuellen Datensegment ein. Die Adressierung erfolgt absolut.
<code>.examine</code>	<code>n</code>	Gibt den Inhalt der durch die Eingabezahl (Bytenummer) festgelegten Hauptspeicherzelle im aktuellen Datensegment aus. Die Adressierung erfolgt absolut.
<code>.in</code>	<code>Kanal_n</code>	Gibt den aktuellen Wert des Ein-/Ausgabe-Kanals aus, der durch die Eingabenummer (0-65535) festgelegt wurde.
<code>.out</code>	<code>Kanal_n1_n2</code>	Weist dem durch <code>n1</code> (0-65535) festgelegten Kanal (Port) die zweite Zahl ( <code>n2</code> ) zu.

## C Dr. Logos Systemvariablen

Tabelle C-1: Dr. Logos Systemvariablen

Variable	Definition
<code>ERRACT</code>	Wenn der Wert von <code>ERRACT TRUE</code> ist, wird bei Fehlererkennung eine Pause eingeleitet.
<code>FALSE</code>	Fester Systemwert.
<code>PE</code>	Wert des Farbstiftzustandes, steht für Radieren ( <code>pe</code> ).
<code>PD</code>	Wert des Farbstiftzustandes, steht für Stiftab ( <code>pd</code> ).
<code>PU</code>	Wert des Farbstiftzustandes, steht für Stifthoch ( <code>pu</code> ).
<code>PX</code>	Wert des Farbstiftzustandes, steht für Umkehrstift ( <code>px</code> ).
<code>REDEFP</code>	Falls der Wert von <code>REDEFP TRUE</code> ist, ist die Neudefinition von
<code>TOPLEVEL</code>	Zustand des Interpretierers mit dem Prompt <code>?</code> . Wenn Dr. Logo ein Fragezeichen ausgibt, befinden sich keine Prozeduren im Stapel und die Ebenennummer ist null. Wenn Sie <code>throw "TOPLEVEL</code> eingeben, werden alle wartenden Prozeduren verlassen.
<code>TRUE</code>	Fester Systemwert.

## D Dr. Logos Systemmerkmale

In Tabelle D-1 sind die Merkmalspaare aufgeführt, die Dr. Logo globalen Variablen, Prozeduren und anderen Objekten zuweist. Ein Merkmalspaar besteht aus einem Merkmalnamen und seinem Wert.

**Tabelle D-1:** Dr. Logos Systemmerkmale

Merkmalname	Merkmalwert
.APV	Zugewiesener Merkmalwert, der Wert einer globalen Variablen.
.DEF	Definition einer Prozedur.
.ENL	Ende einer Prozedurzeile, die durch einen Wagenrücklauf (Return) und Leerzeichen unterbrochen wurde.
.FMT	Anfang einer Prozedurzeile, die durch einen Wagenrücklauf (Return) und Leerzeichen unterbrochen ist.
.PRM	Identifiziert einen Kernbefehl.
.REM	Bemerkung oder Kommentar. (Eine Bemerkung kann auch mit dem Semikolon (;) eingeleitet werden.)

## E Dr. Logos Fehlermeldungen

**Tabelle E-1:** Dr. Logos Fehlermeldungen

2	Number too big Zahl zu groß.
6	(Symbol) is a primitive (Symbol) ist ein Grundwort.
7	Can't find label (Symbol) label (Marke) nicht gefunden.
8	Can't (Symbol) from the editor Im Editor ist (Symbol) nicht möglich.
11	I'm having trouble with the disk Es gibt Probleme mit der Diskette.
12	Disk is full Diskette voll.
13	Can't divide by zero Keine Division durch Null!
15	File (Datei) already exists Datei (Datei) gibt es schon.
17	File (Datei) not found Datei (Datei) nicht gefunden.
21	Can't find catch for (Symbol) Es gibt kein catch für (Symbol).
23	Out of space Zu wenig Hauptspeicher.
25	(Symbol) is not TRUE or FALSE (Symbol) ist weder TRUE noch FALSE.
29	Not enough inputs to (Prozedur) Zu wenig Eingaben für (Prozedur).
32	Too few items in (Liste) Zu wenig Elemente in (Liste).
34	Turtle out of bounds Igel zu weit weg.
35	I don't know how to (Symbol) (Symbol) ist nicht bekannt.
36	(Symbol) has no value (Symbol) hat keinen numerischen Wert.
37	) without ( ) ohne (.

- 38 I don't know what to do with (Symbol)  
 Es ist unklar, was mit (Symbol) gemeint ist.
- 41 (Prozedur) doesn't like (Symbol) as input

572

## F Kommandoliste

Dieser Anhang ist eine Zusammenfassung der Kernbefehle von Dr. Logo. Die Kernbefehle sind nach ihren Funktionen in Gruppen zusammengefaßt. Sofern der Kernbefehl Eingaben erwartet, ist die Form der Eingabe(n) angezeigt.

### Arithmetische Operationen

arctan n  
 cos Grad\_n  
 int n  
 quotient n n (...)  
 random n  
 remainder n n  
 rerandom  
 round n  
 sin Grad\_n  
 + n n (...)  
 - a b  
 \* n n (...)  
 / a b

### Aussagen und Programmsteueroperatoren

bye  
 co <Objekt>  
 go Wort  
 if Aussage Befehlsliste <Befehlsliste>  
 label Wort  
 op Objekt  
 repeat n Befehlsliste  
 run Befehlsliste  
 stop  
 wait n

**Definition von Prozeduren**

```
define Prozname Definitionsliste
end
text Prozname
to Prozname <Eingaben>
```

**Definition von Variablen**

```
local Varname
namep Wort
make Varname Objekt
thing Varname
```

**Disketten**

```
defaultd
setd Laufwerk
```

**Editieren von Prozeduren und Variablen**

```
ed <Name ! Namensliste>
edall
```

**Fehlerbehandlung und Fehlersuche**

```
catch Name Befehlsliste
error
notrace
nowatch <Prozname ! Proznamensliste>
pause
throw Name
trace
watch <Prozname ! Proznamensliste>
```

**Dateibehandlung**

```
changef Neuer_Datname Alter_Datname
dir <Datname>
edf Datname
erasefile Datname
```

```
load Datname
save Datname
```

**Graphische Bewegungen**

```
bk Entfernung_n
fd Entfernung_n
home
ht
lt Grad_n
rt Grad_n
seth Grad_n
setpos Koord_Liste
setx n
sety n
st
towards Koord_Liste
```

**Graphische Zeichenfläche**

```
clean
cs
dirpic <Datname>
dot Koord_Liste
dotc Koord_Liste
erasepic Datname
fence
fill
fs
loadpic Datname
pal Farb_n
pd
pe
pu
px
savepic Datname
setbg Farbe_n
setpal Farbe_n RGB_Liste
setpc Farbe_n
setscrunch n
setsplit n
sf
ss
```

if  
window  
wrap

### Tastatur

keyp  
rc  
rl  
rq

### Logische Operationen

and Aussage Aussage (...)  
not Aussage  
or Aussage Aussage (...)  
= a b  
< a b  
> a b

### Drucker

copyoff  
copyon

### Merkmallisten

glist Merkmal  
gprop Name Merkmal  
plist Name  
pprop Name Merkmal Objekt  
pps  
remprop Name Merkmal

### Tonerzeugung, Lichtgriffel, Steuerknüppel

buttonp Knopf\_n  
ent Parameterliste  
env Parameterliste  
paddle n  
release Kanal\_n  
sound Notenliste

### Textfenster

ct  
cursor  
pr Objekt (...)  
setcursor Koord\_Liste  
show Objekt  
ts  
type Objekt (...)

### Wort- und Listenbearbeitung

ascii Wort  
bf Objekt  
bl Objekt  
char n  
count Objekt  
emptyt Objekt  
first Objekt  
fput Objekt  
last Objekt  
lc Wort  
list Objekt Objekt (...)  
listp Objekt  
lput Objekt Objekt  
memberp Objekt Objekt  
numberp Objekt  
piece n n Objekt  
se Objekt Objekt (...)  
shuffle Liste  
uc Wort  
where  
word Wort Wort  
wordp Objekt

### Arbeitsspeicherverwaltung

er Prozone ! Prozone Liste  
erall  
ern Varname ! Varname Liste  
nodes  
noformat  
po Name ! Namensliste

poall Name : Namensliste

pons  
pops  
pots  
recycle

### Dr. Logos Sonderzeichen

;  
( )  
"  
[]  
!

## G Notenliste

Bei der Klangerzeugung mit dem CPC6128 wird die Tonhöhe durch Eingabe der Tonperiode bestimmt. Der nutzbare Bereich dieser Zahl reicht von 5 für den höchsten Ton bis zu 4095 für den tiefsten Ton. Bei der Eingabe von bestimmten Zahlen für die Tonperiode kommt eine Tonleiter zustande.

Tabelle G-1: Notenliste beim sound-Befehl des CPC6128

Note	Frequenz [Hz]	Tonperiode
C <sub>2</sub>	16,352	3822
C <sub>1</sub> S <sub>2</sub>	17,324	3608
D <sub>2</sub>	18,354	3405
D <sub>1</sub> S <sub>2</sub>	19,445	3214
E <sub>2</sub>	20,602	3034
F <sub>1</sub> S <sub>2</sub>	21,827	2863
F <sub>2</sub>	23,125	2703
G <sub>2</sub>	24,500	2551
G <sub>1</sub> S <sub>2</sub>	25,957	2408
A <sub>2</sub>	27,500	2273
A <sub>1</sub> S <sub>2</sub>	29,135	2145
H <sub>2</sub>	30,868	2025
C <sub>1</sub>	32,703	1911
C <sub>1</sub> S <sub>1</sub>	34,648	1804
D <sub>1</sub>	36,708	1703
D <sub>1</sub> S <sub>1</sub>	38,891	1607
E <sub>1</sub>	41,203	1517
F <sub>1</sub>	43,654	1432
F <sub>1</sub> S <sub>1</sub>	46,249	1351
G <sub>1</sub>	48,999	1276
G <sub>1</sub> S <sub>1</sub>	51,913	1204
A <sub>1</sub>	55,000	1136
A <sub>1</sub> S <sub>1</sub>	58,270	1073
H <sub>1</sub>	61,735	1012

Note	Frequenz [Hz]	Tonperiode
C	65,406	956
CIS	69,296	902
D	73,416	851
DIS	77,782	804
E	82,407	758
F	87,307	716
FIS	92,499	676
G	97,999	638
GIS	103,826	602
A	110,000	568
AIS	116,541	536
H	123,471	506
c	130,813	478
cis	138,591	451
d	146,832	426
dis	155,564	402
e	164,814	379
f	174,614	358
fis	184,997	338
g	195,998	319
gis	207,652	301
a	220,000	284
ais	233,082	268
h	246,942	253
c <sup>1</sup>	261,626	239
cis <sup>1</sup>	277,183	225
d <sup>1</sup>	293,665	213
dis <sup>1</sup>	311,127	201
e <sup>1</sup>	329,628	190
f <sup>1</sup>	349,228	179
fis <sup>1</sup>	369,994	169
g <sup>1</sup>	391,995	159
gis <sup>1</sup>	415,305	150
a <sup>1</sup>	440,000	142
ais <sup>1</sup>	466,164	134
h <sup>1</sup>	493,883	127

mittleres c

Kammerton a

Note	Frequenz [Hz]	Tonperiode
c <sup>2</sup>	523,251	119
cis <sup>2</sup>	554,365	113
d <sup>2</sup>	587,330	106
dis <sup>2</sup>	622,254	100
e <sup>2</sup>	659,255	95
f <sup>2</sup>	698,457	89
fis <sup>2</sup>	739,989	84
g <sup>2</sup>	783,991	80
gis <sup>2</sup>	830,609	75
a <sup>2</sup>	880,000	71
ais <sup>2</sup>	932,328	67
h <sup>2</sup>	987,767	63
c <sup>3</sup>	1046,502	60
cis <sup>3</sup>	1108,731	56
d <sup>3</sup>	1174,659	53
dis <sup>3</sup>	1244,508	50
e <sup>3</sup>	1318,510	47
f <sup>3</sup>	1396,913	45
fis <sup>3</sup>	1479,978	42
g <sup>3</sup>	1567,982	40
gis <sup>3</sup>	1661,219	38
a <sup>3</sup>	1760,000	36
ais <sup>3</sup>	1864,655	34
h <sup>3</sup>	1975,533	32
c <sup>4</sup>	2093,004	30
cis <sup>4</sup>	2217,461	28
d <sup>4</sup>	2349,318	27
dis <sup>4</sup>	2489,016	25
e <sup>4</sup>	2637,021	24
f <sup>4</sup>	2793,826	22
fis <sup>4</sup>	2959,955	21
g <sup>4</sup>	3135,963	20
gis <sup>4</sup>	3322,438	19
a <sup>4</sup>	3520,000	18
ais <sup>4</sup>	3729,310	17
h <sup>4</sup>	3951,066	16

## H Rechnerspezifische Daten

In der folgenden Tabelle sind die Grenzwerte der Text- und Graphikdarstellung für die Standard-Bildschirme des CPC6128 und Joyce zusammengefaßt.

**Tabelle H-1:** Dr. Logos Text- und Graphikdarstellungs-Attribute

Beschreibung	CPC6128	Joyce
Attribute des geteilten Bildschirms		
Standardanzahl der Textzeilen	5	10
Höchste Anzahl der Textzeilen	11	14
Anzahl der Textspalten	40	90
Textschirm-Attribute		
Höchste Anzahl Textfensterspalten	40	90
Höchste Anzahl Textzeilen	24	31
Tabulatorgröße	3	3
Attribute des Grafikbildschirms		
Höchste Anzahl horizontaler Punkte	640	640
Höchste Anzahl vertikaler Punkte	384	529
Höchster Wert in setscrunch	10	10
Niedrigster Wert in setscrunch	1	1
Standardkoordinaten von home	[0 0]	[0 0]
Standard-Zeichenstiftnummer (setpc)	1	1
Höchste Zeichenstiftnummer (setpc)	3	1

### Stichwortverzeichnis

.APV, L-57, L-71  
 .DEF, L-71  
 .ENL, L-72  
 .FMT, L-72  
 .PRM, L-72  
 .REM, L-72

abbrechen, S-14  
 Alt, L-33  
 and, L-60, L-86  
 ändern, S-33  
 Anführungszeichen, L-23, L-27, S-40  
 Anweisung, S-8, S<sup>c</sup>-15  
 Arbeitsdiskette, L-11, S-10  
 Arbeitsspeicher, L-12, L-45, S-9, S-39  
 Arbeitsspeicher organisieren, S-58  
 Arbeitsspeicher überprüfen, S-39, S-42  
 Arbeitsspeicher verwalten, L-73  
 arctan, L-86  
 Argument, L-25  
 ascii, L-28, L-86  
 Ausdruck, L-19, L-30  
 Ausdruck, arithmetischer, L-31  
 Ausgabe, S-51  
 Ausrufezeichen, L-25, L-51, S-13  
 Aussage, L-85

Balkendiagramm, S-61  
 Bedingung erfüllen, S-82  
 beenden, L-17, S-59  
 Begrenzer, L-20, L-21  
 benennen, S-70  
 Betriebssystem, S-9  
 Betriebssystem laden, L-11  
 bf, L-87, S-61  
 Bild speichern L-78  
 Bilddatei, L-80  
 Bildschirm, L-63, S-16

Bildschirm löschen, S-17  
 Bildschirm, geteilter L-45, L-63  
 Bildschirmattribute, L-69  
 Bildschirmauflösung, L-64  
 Bildseitenverhältnis, L-64  
 bk, L-66, L-87, S-18  
 bl, L-88, S-61  
 buttonp, L-89  
 bye, L-89

Caps-Lock, S-11  
 catch, L-90  
 changef, L-79, L-91  
 char, L-91  
 clean, L-67, L-92  
 Clr, S-12, S-34  
 Control, L-33  
 Control-G, S-14  
 Control-S, S-16  
 Control-T, S-16  
 Control-Y, S-13  
 Copy, S-13  
 copyoff, L-92  
 copyon, L-93  
 cos, L-93, L-94  
 count, L-26, L-95  
 cs, L-67, L-93, S-17  
 ct, L-95, S-17  
 Cursor, L-12, L-95

Darstellung, grafische, S-8  
 Darstellung, technisch-wissensch., L-28  
 Datei L-75, S-39  
 Datei laden, S-40  
 Datei löschen, S-41  
 Datei erzeugen, L-80  
 Datei laden, L-15  
 Datei speichern L-14  
 Dateibezeichnung, L-76  
 Dateiname, L-76, S-39  
 Dateiname, mehrdeutiger, L-78

Dateityp, L-76, L-78  
 Dateiverkettung, L-74  
 Datendiskette, S-40, L-11, L-14  
 defaultd, L-17, L-75, L-96  
 define, L-50, L-96  
 Definitionsliste, Format der, L-72  
 Del, S-12, S-34  
 Dezimalpunkt, L-28  
 Dezimalzahl, negative L-28  
 Dezimalzahl, positive L-28  
 Dialog, S-15  
 Dialogverkehr, S-7  
 dir, L-15, L-79, L-97  
 dirpic, L-81, L-97  
 Diskette, L-75, S-39  
 Diskette, leere, S-40  
 Doppelpunkt, S-45, S-70  
 dot, L-66, L-67, L-97  
 dotc, L-66, L-98  
 Dr. Logos Leitfaden, S-8

Ebene, S-87  
 Ebenennummer, L-61  
 ed, L-39, L-99, S-34  
 edall, L-39, L-100  
 edf, L-78, L-100  
 Editier-Kommando, L-33  
 editieren, S-33  
 Editor, S-33  
 Eingabe, S-16, S-45  
 Eingabe, eingebettete, L-85  
 Eingabe, vorangestellte, L-85  
 Eingabeliste, L-55  
 Eingaben wiederholen S-57  
 Eingabeparameter, L-19, L-54  
 Eingabeparameter definieren L-53  
 Eingabezahl, L-55  
 Einrückung, L-51  
 Element ersetzen S-77  
 Element, erstes, S-61  
 Element, letztes, S-61  
 Elemente mischen S-80

empty, L-29, L-101, S-63  
 end, L-39, L-102  
 ent, L-102  
 Enter, S-12  
 env, L-103  
 er, L-80, L-103, S-42  
 erall, L-104, S-42, S-43  
 erasefile, L-79, L-104, S-41  
 erasepic, L-81, L-105  
 ern, L-79, L-105  
 error, L-105  
 Ersatzzeichen, L-77  
 Esc, S-14  
 experimentieren, S-8

FALSE, L-60, S-63  
 Farben, L-68  
 fd, L-66, L-107, S-16  
 Fehler beseitigen S-87  
 Fehler suchen, S-87  
 Fehlersuche, S-88  
 fence, L-68, L-108  
 fill, L-108  
 first, L-108, S-61  
 Fläche, grafische L-63  
 Fließpunktzahl, L-29  
 formatieren, S-9, S-40  
 fput, L-109, S-77  
 Fragezeichen, L-77  
 fs, L-64, L-109  
 Funktionstaste, L-33, L-43, L-45

Genauigkeit, doppelte, L-29  
 glist, L-110  
 go, L-110  
 gprop, L-111, S-74  
 Grafik löschen, L-13  
 Grafik zeichnen, S-8  
 Grafikausgabe, L-63  
 Grafikbildschirm, L-63  
 Grafikprozedur, S-28

Grammatik, L-11  
 Großschreibung, S-30  
 Grundwort, L-12, S-7  
 Grundwortnamen, S-31

Hauptebene, L-20, L-33  
 Hauptspeicher, L-12  
 Hilfe-Meldung, S-13  
 home, L-67, L-112  
 ht, L-113, S-20

if, L-51, L-114, S-59  
 Igel erscheint, S-20  
 Igel links, S-19  
 Igel rechts, S-18  
 Igel rückwärts, S-18  
 Igel verschwindet, S-20  
 Igel bewegen, S-15  
 Informationsübergabe, L-54  
 int, L-115  
 item, L-115

Kernbefehl, L-12, L-83, S-7  
 Kernbefehl, eingebetteter, L-20  
 Kernbefehl, vorangestellter, L-20  
 keyp, L-116  
 Klammer, eckige L-24  
 Kleinschreibung, S-30  
 Knoten, L-73  
 Kommando, S-13, S-15  
 Kommando eingeben, S-16  
 Kommandoprozedur, L-60  
 Kommentar, L-51  
 Kommentare entfernen L-74  
 Konstante, L-52  
 Kontrollzeichen, L-22, L-33, L-45  
 Koordinatenliste, L-66  
 Kopieren, S-11  
 Kurve, S-15

label, L-117  
 laden, L-15, S-40  
 last, L-117, S-61  
 Laufwerk, L-16, L-75, S-9  
 Laufwerkbezeichnung, L-76  
 Laufwerksname, L-84  
 lc, L-118  
 Hauptspeicher, L-12  
 Linie, S-15  
 links, S-19  
 list, L-26, L-118, S-69  
 Liste, L-20, L-25, L-50, L-85, S-23, S-61  
 Liste, literale, L-25  
 Listen, zusammenlegen, S-80  
 Listenelemente manipulieren L-26  
 listp, L-118  
 load, L-79, L-118, S-40  
 loadpic, L-81, L-119  
 local, L-58, L-119  
 Logo-Dateien, L-79  
 löschen, S-39  
 lput, L-119, S-77  
 lt, L-66, L-120, S-19

make, L-29, L-57, L-121, S-70  
 memberp, L-121  
 Mengensymbol, L-83  
 Merkmal, S-74  
 Merkmal anzeigen, S-74  
 Merkmal zuordnen, S-74  
 Merkmalliste, L-56, L-71, S-67, S-73  
 Merkmalname, L-56, L-71, L-84  
 Merkmalpaar, L-56, L-71  
 Merkmalwert, L-56, L-71  
 Mittelpunkt, L-67

Name, L-84  
 namep, L-122  
 nodes, L-73, L-122  
 noformat, L-74, L-122  
 not, L-60, L-123  
 notrace, L-123

nowatch, L-124, S-88  
 numberp, L-124

Objekt, L-23, L-85, S-67  
 op, L-125, S-51  
 Operation, logische, L-60  
 Operator, arithmetischer, L-28  
 Operatorsymbol, L-61  
 or, L-60, L-125  
 Originaldiskette, S-9

paddle, L-127  
 pal, L-127  
 pause, L-38, L-128  
 pd, L-128, S-19  
 pe, L-129  
 Pfeiltasten, S-34  
 piece, L-130  
 Platzhalter, S-45, S-70  
 plist, L-130, S-74  
 po, L-131, S-42  
 poall, L-131  
 pons, L-131, S-72  
 pops, L-131, S-29  
 pots, L-13, L-132, S-42  
 pprop, L-57, L-72, L-132, S-74  
 pps, L-133  
 pr, L-21, L-133  
 Primitive, L-12, S-7  
 Produkt, L-133  
 Programm erstellen, S-51  
 Programm speichern, L-78  
 Programmausführung beobachten, S-87  
 Programmebene, L-61  
 Programmiermethode, gute, L-52  
 Prompt, L-13  
 Prompt, S-10  
 Prozedur, S-7  
 Prozedur ändern, S-33  
 Prozedur beenden, S-59  
 Prozedur benennen, S-27, L-49

Prozedur definieren, L-12, L-19, L-38, L-49  
 Prozedur editieren, S-33  
 Prozedur einfügen, S-27, S-27  
 Prozedur größere, S-53  
 Prozedur interpretieren, L-49  
 Prozedur laden, L-39  
 Prozedur löschen, S-42  
 Prozedur schreiben, S-27, S-28  
 Prozedur sichern, S-39  
 Prozedur strukturieren, L-50  
 Prozedur, Auswertung einer, L-61  
 Prozedur, rekursive, L-55, L-59  
 Prozedur, variable, S-45  
 Prozedur, übersichtliche L-50  
 Prozedurbibliothek, L-19  
 Prozedurdefinition ändern S-33  
 Prozedurdefinition anzeigen, S-42  
 Prozedurebene, S-87  
 Prozedur editieren, L-33, L-38  
 Prozeduren kombinieren, L-19  
 Prozedurname, L-20, L-84, S-27, S-30, S-33, S-67  
 Prozedurnamen auflisten, L-13  
 pu, L-134, S-19  
 px, L-135

quotient, L-136

random, L-137  
 rc, L-137  
 rechts, S-18  
 recycle, L-74, L-138, S-58  
 release, L-138  
 remainder, L-139  
 remprop, L-79, L-139  
 repeat, L-140, S-21, S-23  
 rerandom, L-140  
 Return, S-12  
 rl, L-141, S-82  
 round, L-141

rq, L-25, L-142  
 rt, L-66, L-142, S-18, S-18  
 run, L-142

Satz, anzeigen S-77  
 Satzzeichen, L-11, L-21  
 save, L-14, L-78, L-79, L-144, S-39, S-73  
 savepic, L-78, L-81, L-144  
 Schleife, unendliche S-57  
 Schlüsselwort, L-49  
 Schrägstrich, umgekehrter, L-22  
 schreibgeschützt, S-9  
 se, L-26, L-145  
 Seitenwechsel, L-45  
 Semikolon, L-51  
 setbg, L-68, L-145  
 setcursor, L-146  
 setd, L-17, L-75, L-146  
 seth, L-67, L-147  
 setpal, L-68, L-147  
 setpc, L-68, L-148  
 setpos, L-66, L-67, L-148  
 setscrunch, L-64, L-148  
 setsplit, L-149  
 setx, L-149  
 sety, L-149  
 sf, L-150  
 Shift-Lock, S-11  
 show, L-150  
 shuffle, L-151  
 sichern, S-39  
 Sicherungskopie, L-14, S-10  
 Sicherungskopie erstellen, L-17  
 Sicherungskopie, temporäre, L-71  
 sin, L-151  
 Sonderzeichen, L-77  
 sound, L-151  
 speichern, L-14  
 Speichersäuberung, L-74  
 Speicherung, permanente, L-71  
 Speicherverwaltung, L-71  
 ss, L-64, L-153

st, L-153, S-20  
 Stack, L-61  
 Standard-Laufwerk, L-16  
 Standard-Laufwerk festlegen, L-16  
 Stapel, L-61  
 Stapelspeicherplatz, L-62  
 starten, L-11, S-11  
 stop, L-154  
 Stop, S-14, S-59  
 Summe, L-155  
 Syntax, L-83  
 System-Prompt, S-9  
 Systemdiskette, L-11  
 Systemvariable, L-49

Tabulator, L-34  
 Tastatur, S-11  
 Text, L-156  
 Text eingeben, L-33  
 Text hinzufügen, L-39  
 Text löschen, L-39, S-17  
 Text verändern L-33  
 Text-Bildschirm, S-68  
 Textausgabe, L-63  
 Texteditieren, L-33, L-39  
 Texteditor, S-33  
 Texteingabe, S-28  
 Textfenster, L-64  
 Textformat, S-16  
 Textkoordinaten, L-66  
 Textpuffer, L-63  
 tf, L-156  
 thing, L-157, S-70  
 throw, L-157  
 Tippfehler, L-11  
 Tippfehler korrigieren, L-33  
 Titelzeile, L-53  
 to, L-13, L-158, S-27  
 towards, L-67, L-159  
 trace, L-55, L-160  
 TRUE, L-60, S-59, S-63  
 ts, L-64, L-161

type, L-161  
Typensymbol, L-84

uc, L-162  
Umlaute, S-11

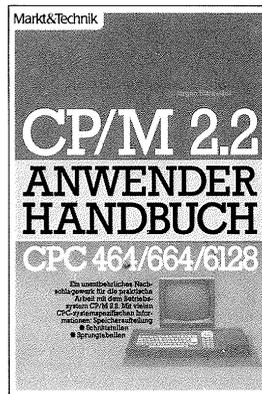
Variable, L-20, L-29, L-52  
Variable erzeugen, L-56  
Variable laden, L-39  
Variable, globale, L-55  
Variable, lokale, L-55  
Variable, Name einer, L-30  
Variable, temporäre, L-57  
Variablen, Inhalt einer, L-30  
Variablenname, L-84  
Vollgrafik, L-45  
Vollgrafik-Bildschirm, L-63  
Volltext, L-45  
Vorgang abbrechen, S-14

wait, L-163  
watch, L-39, L-164, S-87  
watch ausschalten, S-88  
Wertzweisung, indirekte, L-59  
where, L-165  
window, L-165  
word, L-165  
wordp, L-166  
Wort, L-20, L-23, L-84,  
Wort markieren, L-22  
Wort, neues S-28  
Wortelemente manipulieren, L-26  
Wörter eintippen, S-67  
Wörter, S-67  
Wörterliste, S-69  
Wortspiel, S-8, S-77  
wrap, L-68, L-166

x-y-Position, L-67

Zahl, L-20, L-27  
Zehnerpotenz, L-28  
Zeichen löschen, L-34  
Zeichenfläche, L-63, L-66  
Zeichensatz, L-20  
Zeile, L-30  
Zeile ausgeben, S-82  
Zeile löschen, L-34  
Zeile, Ende der, S-13  
Zeilenanfang, L-34  
Zeileneditier-Puffer, L-35  
Zeileneditieren, L-33  
Zeilenende, L-34  
Ziffer, L-27

# Bücher zum Schneider-Computer



J. Hückstädt  
**CP/M-2.2-Anwenderhandbuch**  
**CPC 464/664/6128**  
1985, 212 Seiten

Wenn Sie glücklicher Besitzer eines Schneider-Computers sind und mehr wissen wollen über das leistungsstarke Betriebssystem CP/M 2.2, dann ist dieses Buch genau das Richtige für Sie! Es behandelt CP/M 2.2 nicht nur in seiner allgemeinen Form, wie Sie für sämtliche CP/M-Hardware der CPC-Computer mit ein.

Best-Nr. MT 859  
ISBN 3-89090-204-9  
DM 46,-/sFr. 42,30/6S 358,80



J. Hückstädt  
**CP/M-Plus-**  
**Anwenderhandbuch**  
**CPC 6128/Joyce**  
1986, 256 Seiten

Mit der Verfügbarkeit von CP/M-Plus steht den Besitzern der beliebten Schneider-Heimcomputer der Zugang zu einer der größten Software-Bibliotheken der Welt offen. Allerdings ist es mit der bloßen Gemeinsamkeit des Betriebssystems nicht getan, häufig sind spezielle Anpassungen an den verwendeten Rechner notwendig. Mit dieser gründlichen Darstellung des Betriebssystems sind diese Anpassungen auch für den Laien zu bewältigen. Sie erfahren alles über die Organisation der Dateien, die Grundlagen der Assemblerprogrammierung sowie über den Aufbau von CP/M-Plus.

Best-Nr. MT 90197  
ISBN 3-89090-197-2  
DM 52,-/sFr. 47,80/6S 405,60



H. Tischer  
**Programm-entwicklung**  
**unter CP/M-2.2 auf dem**  
**CPC 464/664**  
1986, 340 Seiten

Theoretische und praktische Teile mit Beispielen in Assembler und Turbo-Pascal vermitteln alle Informationen, die zum selbständigen Entwickeln von CP/M-2.2-Programmen nötig sind. Beschrieben werden sowohl Systemfunktionen als auch die interne Struktur eines vollständigen Standard-CP/M-2.2-Systems. Zwei Kapitel beschäftigen sich nur mit den zusätzlichen Möglichkeiten des CP/Ms der CPC 464/664/6128-Computer. Dabei werden nicht nur die Besonderheiten beim Verwenden eines 3-Diskettenlaufwerkes berücksichtigt, sondern auch die Eigenheiten des Vortex-512-Laufwerkes.

Best-Nr. MT 90209  
ISBN 3-89090-209-X  
DM 52,-/sFr. 47,80/6S 405,60

Markt & Technik-Fachbücher  
erhalten Sie bei  
Ihrem Buchhändler



Markt & Technik Verlag Aktiengesellschaft, Hans-Pinsel-Str. 2, 8013 Haar bei München

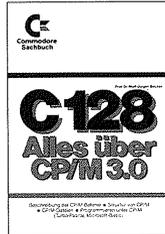
# Bücher zu Commodore 128/128 D



J. Hückstädt  
**CP/M-3.0-Anwender-Handbuch C 128**  
1986, 250 Seiten

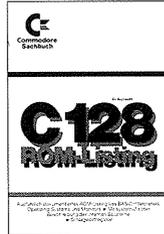
Wenn Sie Ihren Commodore 128 PC schon ganz gut im Griff haben und jetzt so richtig einsteigen wollen in die Möglichkeiten, die das leistungsstarke Betriebssystem CP/M-3.0 bietet, sollten Sie mal in dieses Buch schauen: Es sagt Ihnen alles über den Aufbau einer Datenverarbeitungsanlage, Mikrocomputer, Programmiersprachen und Betriebssysteme im allgemeinen und über das Betriebssystem CP/M speziell auf dem C 128 PC. Ausführliche Beschreibungen der CP/M-Befehle und ihrer Funktionen fehlen ebensowenig wie die umfassende Darstellung der Struktur von CP/M-3.0 auf dem C 128.  
Best-Nr. MT 90196  
ISBN 3-89090-196-4  
DM 52,-/sFr. 47,80/6S 405,60

Markt & Technik-Fachbücher  
erhalten Sie bei  
Ihrem Buchhändler



Prof. Dr. Wolfgang Becker  
**C 128 -  
Alles über CP/M 3.0**  
August 1986, 305 Seiten

Eine fundierte Einführung in die Anwendung des Betriebssystems CP/M 3.0 bzw. CP/M Plus auf dem Commodore 128. Alle installierten Befehle sind mit den wesentlichen Optionen aufgeführt, die Funktionen werden anhand von Beispielen erläutert. Einige CP/M-Software wurde auf den C 128 übertragen. Ein ausführlicher Nachschlageteil enthält systemspezifische Informationen für den Programmierer. Aus dem Inhalt: Einführung in CP/M 3.0, CP/M-Befehle und deren Funktion, Struktur von CP/M auf dem C 128, CP/M-Disketten im C-128-Format, Arbeiten mit CP/M anhand ausgesuchter Softwarebeispiele (Nevada Fortran, Turbo-Pascal).  
Best-Nr. MT 90370  
ISBN 3-89090-370-3  
DM 52,-/sFr. 47,80/6S 405,60



Dr. Ruprecht  
**C 128  
ROM-Listing**  
August 1986, 456 Seiten

Der Besitz eines ROM-Listing ist für diejenigen notwendig, die den C 128 in Maschinensprache programmieren möchten oder ein tieferes Verständnis für ihren Computer erlangen wollen. Dieses Buch beinhaltet ein komplettes, ausführlich dokumentiertes ROM-Listing des Basic-Betriebssystems, des Operating-Systems mit dem 40/80-Zeichen-Editor und des eingebauten Maschinensprache-Monitors. Mit systematischer Beschreibung der internen Bausteine. Sehr nützlich: ein umfassendes Schlagwortregister mit über 100 Wörtern zur Unterstützung eigener Programmieraufgaben.  
Best-Nr. MT 90212  
ISBN 3-89090-212-X  
DM 58,-/sFr. 54,30/6S 460,20



Markt & Technik Verlag Aktiengesellschaft, Hans-Pinsel-Str. 2, 8013 Haar bei München