

北京希望电脑公司 AutoCAD 11.0 技术丛书(之四)

AutoCAD 11.0

計算機繪圖軟件包

AutoLISP 程序員參考手冊



海洋出版社

北京希望电脑公司 AutoCAD 11.0 技术丛书(之四)

AutoCAD 11.0

计算机绘图软件包

主 编 李 振 格

AutoLISP 程序员参考手册

编 者 章 忆 文
审 校 田 德 中

海 洋 出 版 社

内 容 提 要

AutoCAD 11.0 是计算机辅助设计、辅助绘图的通用软件包,是一套功能极强的工具。AutoCAD 11.0 具有极强的 3 维绘图功能。

《AutoLISP 程序员参考手册》是 autoCAD 系列丛书之一,描述了 AutoCAD 中的高级设计语言 AutoLISP 的功能、安装、所有内部子程序的用法。在说明子程序的用法时全都采用了例子。

本书结构严谨,层次清晰,文字经过认真编辑,通顺、简练、易懂。

本书是 AutoCAD 11.0 开发应用人员必备之书。

本丛书分批出版,欲购此丛书及其汉化软件者请直接与北京 8721 信箱联系,电话 2562329。

北京电脑公司 AutoCAD 11.0 系列丛书 之 四

AutoLISP 程序员参考手册

主 编 李 振 格

编 者 章 忆 文

责任编辑 阎 世 尊

海洋出版社出版(北京市复兴门外大街 1 号)

双青印刷厂印刷

开本: 787x1092毫米 1/16 印张: 10.0 字数: 243千字

1991年5月第1版

1991年5月第1次印刷

印数: 1—3000

ISBN7-5027-2062-6/TP·29 定价: 9.0元

北京希望电脑公司推出 AutoCAD 11.0 系列丛书

- AutoCAD用户参考手册(之一)
- AutoCAD安装和性能指南(之二)
- AutoCAD自学手册 (之三)
- AutoLISP程序员参考手册(本书)(之四)
- ADS程序员参考手册(之五)
- AutoCAD高级造型扩展参考手册(之六)
- AutoCAD实用程序和C(Microsoft C & Turbo C)函数包集锦(之七)

以上为联机手册, 是AutoCAD应用开发人员必备之丛书

下面是以开发应用为主线的AutoCAD丛书

- AutoCAD开发应用教程(之八)
- AutoCAD编程参考手册(之九)
- AutoCAD 1.0~11.0百科全书(上、下册)(之十)

AutoCAD 11.0的西文和全汉化软件请与北京8721信箱资料部联系, 邮编100080, 电话2562329。

目 录

第零章 简介	1
0.1 相关的产品	1
0.2 为什么要采用 LISP?	1
0.3 AutoLISP 中的数据类型	2
0.4 AutoLISP 求值程序	2
0.5 语法规则	3
0.6 符号约定	4
0.7 最新的修改和补充	5
0.8 出错处理	5
第一章 AutoLISP 的安装	6
1.1 标准的 AutoLISP	6
1.1.1 配置	6
1.1.2 环境变量	6
1.2 扩展 AutoLISP	6
1.2.1 配置	7
1.2.2 环境变量	7
1.2.3 使用扩展 AutoLISP	7
1.3 文件 acad.lsp 的自动函数定义	7
第二章 AutoLISP 基础知识指导	9
2.1 预备知识	9
2.2 目标	9
2.3 开始	10
2.4 开始输入	11
2.5 方位输入	13
2.6 绘制砖块	15
2.7 把命令增补到 AutoCAD	17
2.8 清理显示屏幕	20
2.9 总结	22
第三章 AutoLISP 函数	24
3.1 AutoLISP 函数分类	24
3.1.1 (+ 数 数)	24
3.1.2 (- 数 [数] ...)	24
3.1.3 (* 数 [数] ...)	24

3.1.4	(/数 [数 ...])	25
3.1.5	(= 原子 原子 ...)	25
3.1.6	(/= 原子 原子 ...)	25
3.1.7	(< 原子 原子 ...)	25
3.1.8	(<= 原子 原子 ...)	25
3.1.9	(> 原子 原子 ...)	26
3.1.10	(>= 原子 原子 ...)	26
3.1.11	(~ 数)	26
3.1.12	(1+ 数)	26
3.1.13	(1- 数)	26
3.1.14	(abs 数)	27
3.1.15	(ads)	27
3.1.16	(and 表达式 ...)	27
3.1.17	(angle pt1 pt2)	27
3.1.18	(angtos 角 [方式 [精度]])	27
3.1.19	(append expr ...)	28
3.1.20	(apply 函数表)	28
3.1.21	(ascii 字符串)	28
3.1.22	(assoc 项 关联表)	29
3.1.23	(atan num1 [num2])	29
3.1.24	(atof 字符串)	29
3.1.25	(atoi 字符串)	29
3.1.26	(atom 项)	30
3.1.27	(Boole func int1 inte ...)	30
3.1.28	(boundp atom)	31
3.1.29	(caar 表), (cadr 表), (caddr 表), (cadar 表), 等等	31
3.1.30	(car 表)	31
3.1.31	(cdr 表)	32
3.1.32	(chr 整数)	32
3.1.33	(close file-desc)	32
3.1.34	(command [args] ...)	32
3.1.35	(cond (测试 1 结果 ...) ...)	33
3.1.36	(cons new-first-element list)	34
3.1.37	(cos 角度)	34
3.1.38	(cvunit value from to)	34
3.1.39	(defun sym 变元表 expr ...)	35
3.1.39.1	函数库和自动装入	36
3.1.39.2	C:XXX 函数——向 AutoCAD 增加命令	36
3.1.39.3	S::XX 函数——自动执行	37
3.1.39.4	(distance pt1 pt2)	37

3.1.39.5	(eq expr1 expr2)·····	37
3.1.39.6	(equal expr1 expr2 [误差量])·····	38
3.1.40	(eval expr)·····	38
3.1.41	(exp 数)·····	38
3.1.42	(expt 底数幂)·····	39
3.1.43	(findfile 文件名)·····	39
3.1.44	(fix 数)·····	39
3.1.45	(float 数)·····	39
3.1.46	(foreach name list expr ...)·····	39
3.1.47	(gcd num1 num2)·····	40
3.1.48	(getangle [pt] [prompt])·····	40
3.1.49	(getcorner pt [prompt])·····	41
3.1.50	(getdist [pt] [prompt])·····	41
3.1.51	(getrer variable-name)·····	41
3.1.52	(getint [prompt])·····	42
3.1.53	(getkword [prompt])·····	42
3.1.54	(getorient [pt] [prompt])·····	42
3.1.55	(getpoint [pt] [prompt])·····	43
3.1.56	(getreal [prompt])·····	43
3.1.57	(getstring [cr] [prompt])·····	43
3.1.58	(getvar varname)·····	44
3.1.59	(graphscr)·····	44
3.1.61	(initget [bits] [string])·····	44
3.1.62	(inters pt1 pt2 pt3 pt4 [onseg])·····	46
3.1.63	(itoa int)·····	47
3.1.64	(lambda arguments expr ...)·····	47
3.1.65	(last list)·····	47
3.1.66	(length list)·····	48
3.1.67	(list expr ...)·····	48
3.1.68	(listp item)·····	48
3.1.69	(load filename [onfailure])·····	48
3.1.70	(log number)·····	49
3.1.71	(log and number number ...)·····	49
3.1.72	(logior integer ...)·····	49
3.1.73	(lsh num1 numbits)·····	49
3.1.74	(mapcar function list1 ... listn)·····	50
3.1.75	(max number number ...)·····	50
3.1.76	(member expr list)·····	51
3.1.77	(menucmd string)·····	51
3.1.78	(min number number)·····	51

3.1.79	(minusp item).....	52
3.1.80	(not item).....	52
3.1.81	(nth n list).....	52
3.1.82	(null item).....	52
3.1.83	(numberp item).....	53
3.1.84	(open <filename> <mode>).....	53
3.1.85	(or expr).....	54
3.1.86	(osnap pt mode-string).....	54
3.1.87	pi.....	54
3.1.88	(polar pt angle distance).....	54
3.1.89	(prin1 [expr [file-desc]]).....	54
3.1.90	(princ[expr[file-desc]]).....	55
3.1.91	(print[expr[file-desc]]).....	55
3.1.92	(progn expr...).....	55
3.1.93	(prompt msg).....	56
3.1.94	(quote expr).....	56
3.1.95	(read string).....	56
3.1.96	(read-char[file-desc]).....	57
3.1.97	(read-line[file-desc]).....	57
3.1.98	(redraw[ename[mode]]).....	57
3.1.99	(rem num1 num2...).....	58
3.1.100	(repeat number expr...).....	58
3.1.101	(reverse list).....	58
3.1.102	(rtos number[mode[precision]]).....	58
3.1.103	(set sym expr).....	59
3.1.104	(setvar varname value).....	60
3.1.105	(sin angle).....	60
3.1.106	(sqrt number).....	60
3.1.107	(strcase string[which]).....	61
3.1.108	(strcat string1[string2]...).....	61
3.1.109	(Strlen[string]).....	61
3.1.110	(subst newitem olditem list).....	61
3.1.111	(substr< string> <start> [length]).....	62
3.1.112	(terpri).....	62
3.1.113	(textpage).....	62
3.1.114	(textscr).....	62
3.1.115	(trace function...).....	62
3.1.116	(trans pt from to [disp]).....	63
3.1.117	(type item).....	64
3.1.118	(untrace function...).....	65

3.1.119	(ver).....	65
3.1.120	(VPORTS).....	66
3.1.121	(Wcmatch string pattern).....	66
3.1.122	(while textexpr expr...).....	67
3.1.123	(write-char num [file-desc]).....	68
3.1.124	(write-line string[file-desc]).....	68
3.1.125	(xload application).....	68
3.1.126	(xunload application).....	68
3.1.127	(zerop item).....	69
3.1.128	(*error* string).....	69
第四章	实体和设备的访问	70
4.1	特殊数据类型.....	70
4.2	选择集操作函数.....	70
4.2.1	(ssget [mode] [pt1 [pt2]]).....	70
4.2.2	(sslength ss).....	72
4.2.3	(ssname ss index).....	72
4.2.4	(ssadd [ename[ss]]).....	72
4.2.5	(ssdel ename ss).....	73
4.2.6	(ssmemb ename ss).....	73
4.3	实体 名称 函数.....	73
4.3.1	(entext [ename]).....	73
4.3.2	(entlast).....	74
4.3.3	(entsel [prompt]).....	74
4.3.4	(nentsel [prompts]).....	74
4.3.5	(handent handle).....	77
4.4	实体数据函数.....	77
4.4.1	(entdel ename).....	77
4.4.2	(entget ename[applist]).....	78
4.4.3	(entmake [elist]).....	80
4.4.4	(entmod elist).....	82
4.4.5	(entupd ename).....	83
4.4.6	限制.....	84
4.5	将实体名和选择集与 AutoCAD 一起使用.....	84
4.6	对处理曲线拟合和样条拟合多义线的说明.....	84
4.7	扩展实体数据函数.....	85
4.7.1	(regapp application).....	85
4.7.2	(xdroom ename).....	85
4.7.3	(xdsize list).....	86
4.8	符号存取函数.....	88

4.8.1	(tblnext table-name[rewind])	88
4.8.2	(tblsearch table-name symbol[setnext])	89
4.9	对图形屏幕和输入设备的存取	89
4.9.1	(grclear)	90
4.9.2	(grdraw from to color [highlight])	90
4.9.3	(grtext [box text[highlight]])	90
4.9.4	(grread [itrace])	91
第五章	内存管理	92
5.1	AutoLISP 的内存调整方法	92
5.1.1	环境变量影响 AutoLISP 内存的使用	92
5.2	恢复结点空间	95
5.3	虚拟函数页管理程序	96
5.4	技术性注释	96
5.4.1	节点空间	96
5.4.2	字符串空间	97
5.4.3	符号存贮	97
5.4.4	人为分配	97
5.4.5	内存统计	98
5.4.6	虚拟函数页或管理页表	98
附录 A	提供的程序	100
A.1	装入程序	100
A.2	运行程序	100
A.3	距离与角度	101
A.4	二维绘图程序	101
A.4.1	Asctext.lsp 插入 ASCII 文本文件	101
A.4.2	Cbtext.lsp 修改文本	102
A.4.2.1	高度选项	103
A.4.2.2	校准选项	103
A.4.2.3	位置选项	103
A.4.2.4	旋转选项	103
A.4.2.5	字型选择	104
A.4.2.6	文本选项	104
A.4.2.7	取消选项	105
A.4.2.8	宽度选项	105
A.4.3	Cl.lsp 构造中心线	105
A.4.4	Dline.lsp 绘制双线	106
A.4.4.1	Break 选项	106
A.4.4.2	caps 选项	106

A.4.4.3	Dragline 选项	107
A.4.4.4	Offset 选项	107
A.4.4.5	Snap 选项	107
A.4.4.6	Undo 选项	108
A.4.4.7	Width 选项	108
A.4.5	Ptext.lsp 创建并编辑文本段	108
A.4.5.1	Center 选项	109
A.4.5.2	Edit 选项	109
A.4.5.3	Fit 选项	109
A.4.5.4	Right 选项	109
A.4.5.5	Slack 选项	109
A.4.5.6	?选项	109
A.4.5.7	<Start point>选项	110
A.4.5.8	Control 字符	110
A.4.6	Rectang.lsp 绘制一个正方形或一个矩形	110
A.4.7	Rpoly.lsp 定义多边形	111
A.4.8	Spiral.lsp 建立一个 2 维螺旋线	111
A.5	3D 草图程序	111
A.5.1	3D.lsp 创建 3D(Wireframe)对象	111
A.5.1.1	Box 创建一个 3D 的立方体	112
A.5.1.2	Cone(圆锥)	113
A.5.1.3	Dome 和 Dish 创建一多边形的网格半球	113
A.5.1.4	Mesh 创建一样本平面网格	114
A.5.1.5	Pyramid	115
A.5.1.6	Sphere	116
A.5.1.7	Torus	117
A.5.1.8	Wedge	117
A.5.2	3Darray.lsp 建立 3D 长方体和锥体阵列	118
A.5.3	Axrot.lsp 绕一轴旋转实体	120
A.5.4	Chface.lsp 移动 3D 表面顶点	120
A.5.4.1	1, 2, 3 或 4 选项	121
A.5.4.2	Undo 选项	121
A.5.4.3	Display 选项	121
A.5.4.4	<select vertex> 选项	121
A.5.5	Edge.lsp 修改 3D 表面边的可见性。	121
A.5.6	Mface.lsp 创建 Pface 实体	121
A.5.6.1	Layer 选项	122
A.5.6.2	Color 选项	122
A.5.6.3	<Select Vertex> 选项	122
A.5.7	Project.lsp 把 3D 模型投影到一个平面。	122

A.5.7.1	project1	122
A.5.7.2	Project2	123
A.6	实用子程序	123
A.6.1	Alias.lsp 显示命令别名表	123
A.6.2	Attreddef.lsp 更新并重定义属性	123
A.6.3	Calc.lsp 联机计算器	123
A.6.3.1	Calc 选项	124
A.6.3.2	Mem 选项	124
A.5.6.3	Trig 选项	125
A.6.4	chblock.lsp 分别修改一个块的 X 轴、Y 轴和 Z 轴的比例尺。	126
A.6.4.1	Insertion point 选项	126
A.6.4.2	Rotation 选项	126
A.6.4.3	Scale 选项	126
A.6.4.4	<Exit> 选项	126
A.6.5	Dellayer.lsp 删除一个层中的所有实体	126
A.6.6	End.lsp 改进 End 命令	127
A.6.7	Ep.lsp 输入一个点	127
A.6.7.1	Exit 选项	127
A.6.7.2	World 选项	127
A.6.7.3	set lastpoint 选项	127
A.6.7.4	Absolute to UCS origin 选项	127
A.6.7.5	(Relative to lastpoint> 选项	127
A.6.9	Fcopy.lsp 复制一个文本文件到另一文本文件	128
A.6.10	Fplot.lsp 用双变量之函数绘制	129
A.6.11	Fprint.lsp 在屏幕上显示文本文件	130
A.6.12	Lload.lsp 表和 ADS 应用程序加载程序	130
A.6.12.1	Add 选项	131
A.6.12.2	remove an entry 选项	131
A.6.12.3	<Number to load> 选项	131
A.6.13	Mvsetup.lsp 设置画图之配置	131
A.6.13.1	TILEMODE=1 (ON)	132
A.6.13.2	TILEMODE=0 (Off)	132
A.6.14	Sgr.lsp 计算平方根	135
A.6.15	Ssx.lsp 一个简便的(ssget "X")函数	135
A.6.16	Tables.lsp 显示/分类符号表	136
A.6.17	Xdata.lsp 罗列或添加扩展实体数据	136
A.6.17.1	Xdata	137
A.6.17.2	Xdlist	138
A.6.18	Xplode.lsp 修改分解命令	139
A.6.18.1	All 选项	139

A.6.18.2	Color 选项	139
A.6.18.3	Layer 选项	140
A.6.18.4	LType 选项	140
A.6.18.5	Inherit from Parent Block 选项	140
A.6.18.6	<Explode> 选项	140
A.6.19	Xrefclip.lsp 插入和截取外部引用块	140
A.7	AutoShade 子程序	141
A.7.1	Ashade.lsp 更新 AutoShade 文件	141
附录 B	错误信息	142
B.1	用户程序错误	142
B.2	内部错误	146
附录 C	错误代码	148

第零章 简介

AutoLISP 是嵌入于 AutoCAD 软件包的 LISP 编程语言。AutoLISP 让用户和 AutoLISP 开发者以适合图形应用的强大的高级语言来编写宏程序和函数。LISP 既易学易用, 又非常灵活。

用户不必知道在 AutoLISP 中如何编写能有效适用于 AutoCAD 的程序。如果你对于计算机编程没有经验, 可以首先通过第一章的研读来学习如何安装 AutoLISP, 然后使用附录 A 中所描述的提供的给定程序。它将使你学习使用那些调用 AutoLISP 的具有大量 discipline specific 规则说明的 AutoCAD 菜单和应用程序。如果你喜欢编程, 那么你就应当研读本手册的其余部分, 以便学习怎样用 AutoLISP 把通用的 AutoCAD 软件包转化为适合专业需要的更强有力的工具。

本手册是一个参考手册而不是一个 LISP 编程教科书。尽管在第二章中有如何使用 AutoLISP 的实例, 我们仍建议用户先看一些有关 LISP 的教科书, 以使能更快学会 AutoLISP 编程语言。推荐的参考书目有由 Winston 和 Horn 编写的《LISP》(第二版)和由 Tony Hasemer 编写的《Looking at LISP》, 这两本书均由 Addison_Wesley 出版社出版。LISP 是一种具有很多语系的语言, 它包括 MacLISP, InterLISP, ZetaLISP 和 CommonLISP。AutoLISP 采用了和 CommonLISP 最相近的语法和习惯约定, 但 AutoLISP 只是一个很小的子集, 并且 AutoCAD 还为 AutoLISP 增设了许多专用函数。本参考手册列出了 AutoLISP 的所有函数及其使用方法。

0.1 相关的产品

Autodesk 软件开发工具(SDK)包括提高 ADS 生产率的工具, 提高 AutoLISP 相关产品的执行的工具, 以及配制和设置 AutoCAD 以适合不同的应用的工具。SDK 主要设计用来为有经验的编程者和专业的开发者的高级应用: Autodesk 并不为 SDK 提供技术支持。然而, 支持信息可从 CompuServe 网络上的 Autodesk 软件开发协会获得。

可以通过邮递购买 SDK。

0.2 为什么要采用 LISP?

我们选择 LISP 为 AutoCAD 的第一种应用接口语言, 有如下一些原因:

- LISP 在所有的编程语言中是最易于学习和掌握的。
- LISP 也是我们选作用于研究和开发人工智能和专家系统的工具。
- 由于 LISP 本身具有特别简洁的语法, 所以编写 LISP 解释程序变得相当简易, 其程序尺寸也非常小。
- LISP 解释程序最适于那种专用于设计过程中的自由式的交互操作。
- LISP 擅长处理具有不同存储容量的各类数据对象, 而这正是 CAD 系统如 AutoCAD 所需要的。

随着版本 11 的 AutoCAD 开发系统(ADS)的应用,应用程序也可以用其它语言来编写。然而, Autodesk 将保证长期支持 AutoLISP, 并且 ADS 作为 AutoLISP 的补充, 以供用户选用。

0.3 AutoLISP 中的数据类型

AutoLISP 支持几种数据类型, 它们是:

- 表
- 符号
- 字符串
- 实型数
- 整型数
- 文件句柄
- AutoCAD 的实体名
- AutoCAD 的选择集
- 子程序(内部函数)
- 外部子程序(ADS 函数)

在 PC-DOS, MS-DOS 及 OS/2 1.1, 1.2 系统中, 整数为 16 位带符号数值, 其范围是从 -32768 到 +32767。在 32 位工作站上和具有 phar Lap 扩展内存的 DOS 机中, AutoLISP 的整数为 32 位带符号数值, 其范围是从 -2,147,483,648 到 +2,147,483,647。但 AutoLISP 和 AutoCAD 之间的整数传输被限制在 16 位数值的范围内。

实数用双精度浮点数表示, 并且至少有 15 位有效精度。字符串可以是任意长度; 它们的存贮空间是动态分配的。虽然输入字符串限制在 100 个字符, 但可用 `strcat` 函数来生成无长度限制的字符串。

AutoLISP 的几个内部函数提供了二维和三维图形编程应用的基础。图形坐标的表示规则如下:

二维点 以两个实数(x, y)的表形式表示, 如:(3.4 7.52),
 第一个值是x坐标, 第二个值是y坐标。

三维点 以三个实数(x, y, z)的表形式表示, 如:(3.4 7.52 1.0)
 第一个值是x坐标, 第二个值是y坐标, 第三个值是z坐标。

当 AutoCAD 要求某种类型的输入(一个点或一个比例因子)时, 则可使用该类型的 AutoLISP 表达式或可返回该类型结果的 AutoLISP 函数来提供所需的值。

0.4 AutoLISP 求值程序

每个 LISP 解释程序的核心就是其求值程序。求值程序读入用户的输入行, 对它进行计算, 并返回其结果。下面是 AutoLISP 的求值过程:

- 整型数, 实型数, 字符串, 文件句柄和子程序, 以它们本身的值作为结果。
- 符号以它们当前的约束值作为计算结果。
- 表是根据它的第一个元素进行计算的。
- 如果第一个元素计算的结果是一个表(或空), 那么整个表就被假定为函数定义, 函数通过使用表中以剩余元素为变元的值来进行计算。
- 如果第一个元素的计算结果是一个内部函数(子程序)名, 那么表中剩余元素作为形式变元传送给子程序, 并由子程序进行计算。

如果用输入一个 AutoLISP 表达式来响应 AutoCAD 的命令提示, 则 AutoLISP 计算该表达式的值并打印出计算结果。然后 AutoCAD 的“Command:”提示重新出现。当打印实数时, AutoLISP 将只显示到 6 位有效数字。

如果键入或从文件中读入一个错误的表达式, AutoLISP 可能显示下列信息:

n>

这里的 n 是一个整数, 它指示出没有封闭的左括号的级数。如果这种提示出现, 必须要键入 n 个右括号, 以退出这种状态。一种常见的错误是漏掉了正文字符串中闭合的双引号, 在这种情况下, 右括号将被解释为括起来的字符, 键入 n 个右括号。要纠正这种错误, 先要键入 ctrl+C 废除该函数然后重新正确地键入(先键入一个双引号然后键入 n 个右括号)。

0.5 词法规则

AutoLISP 的输入可以采用若干种形式。可以在运行 AutoCAD 时从键盘键入, 可以从 ASCII 文件中读入, 也可以从字符串变量中读入。所有的这些形式都必须遵守一定的约定:

- 符号名可以由除下列字符之外的所有可打印字符序列组成:
() . ' " ;
- 表达式可扩展到多行
- 下列字符终止一个符号名或数字常量。
() ' " ; (space) (end of line)
- 符号之间留多个空格等价于留一个空格。虽然行首缩进不是必需的, 但使用行首缩进可使用户函数的层次结构更加清晰。
- 在 AutoLISP 中, 符号和函数名(子程序)的大小写等效。
- 整型常数之前可以加一个任选的“+”或“-”字符。如上所述, 其值域是从-32,768 到 +32,767。
- 实型常数由一个或多个数字, 后面跟着小数点, 小数点后再跟着一个或多个数字组成。例如, “.4”并不被认作是实数, “0.4”才是正确的写法。同样, “.5”不能被认作是实数, “5.0”才是正确的写法。实数可以用科学计数法表示, 即:它可以有一个任选的“e”或“E”, 其后是指数部分。
- 文字字符串是由双引号引起来的字符序列。在引号中的字符串里, 使用反斜杠

(\), 以使得某些控制字符可被正确识别。目前这些控制符表示成:

\\	表示字符\
\"	表示字符"
\e	表示escape (ESC)
\n	表示换行
\r	表示回车
\t	表示制表符
\nnn	表示其八进制码为nnn的字符

例如, 下面的文字串将在新行上发出提示:

```
(prompt "\n Enter first point:)
```

- 单引号可以作为 quote 函数的缩写, 因此:

```
'foo
```

等价于

```
(quote foo)
```

- 从磁盘文件中装入的 AutoLISP 文件可包含注解。注解要以分号开头, 分号后一直到行尾都为注解部分。例如:

```
; This entire line is a comment
```

```
(setq area (* pi r r) ) ; compute area of circle
```

0.6 符号约定

本参考手册使用了一定的约定来描述函数的有关特性。例如:

```
(moo string number ...)
```

函数名按所输入的显示。在函数名之后尖括号中的项表示所带的变元数目和类型。

在这个例子中, 函数 moo 要求有两个变元: 一个字符串和一个数。省略号("...")表示可在函数后面加额外的数字变元。当调用函数时, 不能包括有尖括号和省略号。按照上面所示的 moo 函数的调用格式, 下面对 moo 函数的调用都是有效的:

```
(moo "Hello" 5)
```

```
(moo "Hi" 1 2 3)
```

下面的例子不符合所述的格式, 将会出错:

```
(moo 1 2 3) (第一变元必须是一个字符串)
```

```
(moo "Hello") (至少应有一个数字变元)
```

```
(moo "do" '(1 2)) (第二个变元必须是一个数, 而不是一个表)
```

当任选的变元只能出现一次而不能重复出现时, 这个变元就用方括号("[])"括起来, 如:

```
(foo string [number])
```