

计算机图形与图像丛书



陈文贤 编著

深入剖析 AutoLISP

学苑出版社

计算机图形与图像丛书

深入剖析 AutoLISP

陈文贤	编著
田红霞 谷 鸥	等编写
甘 特 沈 晓	审校

学 苑 出 版 社

1 9 9 4

(京) 新登字 151 号

内 容 提 要

本书详细介绍了 AutoLISP 的各种功能。其中包括怎样建立用户的系统, 使用子程序的好处, 处理实体, 使用图形库表格定义, 读写屏幕及输入设备的数据, 使用实体属性作为工具或作为报表, 以参数方式建立图形, 并且讨论了 AutoLISP 与文件、设备、其它软件、操作系统的连接, 还包括设计 3D 图形程序, 利用 Dxf 文件修改图形等内容。

本书内容全面, 叙述清晰, 使用查阅方便, 是从事 AutoLISP 程序设计人员的极为有用的工具书, 又是计算机应用人员、大专院校师生必备的参考书。

欲购本书者, 请与北京 8721 信箱联系, 邮政编码 100080, 电话 2562329。

版 权 声 明

本书繁体字中文版原名为《剖析 AutoLISP》, 由松岗电脑图书资料股份有限公司出版, 版权归松岗公司所有。本书简体字中文版版权由松岗公司授予北京希望电脑公司, 由北京希望电脑公司和学苑出版社独家出版、发行。未经出版者书面许可, 本书的任何部分不得以任何形式或任何手段复制或传播。

计算机图形与图像丛书

深入剖析 AutoLISP

编 著: 陈文贤
改 编: 田红霞 谷 鸥
审 校: 甘 特 沈 晓
责任编辑: 徐建军
出版发行: 学苑出版社 邮政编码: 100032
社 址: 北京市西城区成方街 33 号
印 刷: 双青印刷厂
开 本: 787×1092 1/16
印 张: 25.875 字数: 602 千字
印 数: 1-5000 册
版 次: 1994 年 2 月北京第 1 版第 1 次
ISBN 7-5077-0884-5/TP·26
本册定价: 34.00 元

学苑版图书印、装错误可随时退换

目 录

第零章	>> AutoLisp 函数中英文参考总表及实例	1
第一章	开始学习 AutoCAD 系统	47
1.1	必需具备的目录	48
1.2	文本编辑器(texteditor)	49
1.3	DOS 启动环境设置	49
1.4	AutoCAD 系统程序与支持文件	50
1.5	安装设置 AutoCAD 工作环境	51
1.6	用 DOS 批处理文件启动 AutoCAD	52
1.7	原型图形文件	53
1.8	图层取名编名法	54
第二章	AutoLISP 的基本概念与程序	56
2.1	AutoLISP 的好处	57
2.2	如何写一个简单的 AutoLISP 表达式	59
2.3	设置 AutoLISP 变量值	59
2.4	有关一个菜单项的举例说明	60
2.5	AutoLISP 将输入命令送给 AutoCAD 执行	61
2.6	由用户获取输入的函数	61
2.7	注意要点	62
2.8	一个简单的 AutoLISP 程序的开发	62
2.9	Acad.lsp: 自动加载的 AutoLISP 文件	67
第三章	AutoLISP 的基本元素	69
3.1	作为各种输入用的 get 函数	70
3.2	使用 AutoLISP 的数学函数	71
3.3	字符串函数	72
3.4	AutoLISP 的 lists(序列)是如何工作的	72
3.5	定义自己的 AutoLISP 函数	73
第四章	了解 AutoLISP 的程序流程(结构化 AutoLISP 程序)	76
4.1	关系运算函数	77
4.2	条件函数	78
4.3	程序循环	79
4.4	构成复合语句的 Progn 函数	80
4.5	程序设计环境	80
4.6	有效率地使用变量名	82
4.7	子程序	83

4.8 递归程序设计	84
第五章 与 AutoLISP 联系建立用户接口工具	86
5.1 决定数据类型	87
5.2 字符串的处理	89
5.3 转换数据类型	92
5.4 Atext 程序中应用转换及提示格式	94
5.5 开发用户接口函数	97
5.6 用 initget 函数控制输入	99
5.7 将所有用户接口集成成工具箱	104
第六章 存取图形数据库	107
6.1 AutoCAD 的实体	108
6.2 实体选择集(Entity Selection sets)	111
6.3 将实体选择集工具函数加入 Aplate.lsp 程序中	113
6.4 实体集选择模式(Optional Selection set modes)	114
6.5 开发实体选择工具	115
6.6 实体数据(Entity Data)	115
6.7 实体关连序列与点分数据对	116
6.8 DXF 的组码与 Autolist	117
6.9 缺省实体属性	120
6.10 在 CSCALE 程序中应用实体存取功能	120
6.11 修改与更正实体数据	123
6.12 复杂实体 polyline(组合线)及 inserts(插入实体)	125
6.13 实体句柄永久的实体名称	129
6.14 摘要	129
第七章 AutoCAD 的表数据	131
7.1 符号表	132
7.2 命名图层、字体与视图	135
7.3 探讨块段	137
7.4 命名性质: 线型及其它	146
7.5 视区 VPORT 表及用户坐标系 UCS	151
7.6 应用 vports 函数来改变视区设置	154
第八章 AutoLISP 输出输入(I/O)读写数据	156
8.1 数据读写	151
8.2 一般的文件处理(General File Handling)	160
8.3 写数据到打印机或其它设备	161
8.4 测试文件及路径的 AutoLISP 程序	163
8.5 应用文件处理制作一个 pattern.lsp 剖面图案生成器	171
8.6 屏幕格式化 ANSI 码	178
8.7 格式化文件供外部处理	180

8.8	在 Refdwg 程序中应用数据文件的存取	181
8.9	摘要	187
第九章	AutoLISP 与外部设备的联接(图形接口式的数据 I/O)	189
9.1	用 Grtext 在屏幕上绘动态字符串	190
9.2	以 grdraw 函数绘出暂时性线向量	193
9.3	建立动态预视命令 pview	194
9.4	用 grread 取得设备的输入(device input)	197
9.5	应用 grread 到 etext.lsp 程序(AutoCAD 文字处理器)	198
9.6	利用 grread 函数, 连续式追踪光标坐标值	201
9.7	在 ddraw.lsp 程序中应用 grread 的追踪能力	202
9.8	摘要	205
第十章	结合 AutoLISP 与 Acad.menu 功能选项	
	(以一个 ISO 尺寸标注系统为例)	206
10.1	宏命令与菜单的复习	207
10.2	设计简洁的宏命令菜单	209
10.3	同一屏幕菜单项上多个值的选择更替设置	211
10.4	在等轴测尺寸标注中应用动态菜单切换	214
10.5	建立轴测尺寸标注函数	215
10.6	建立等轴尺寸标注屏幕菜单	216
第十一章	利用属性作为自动化数据处理工具	223
11.1	AutoCAD 属性数据的处理	224
11.2	用属性来控制文本绘制	228
11.4	更新属性	233
11.5	属性编辑自动化	234
11.6	用 AutoLISP 程序作绘图时间登录工作	235
11.7	建立一个历次图形文件修改摘要登记系统	237
11.8	利用 Update.lsp 程序作修改块的自动插入工作	238
11.9	将参数数据存储在属性中	244
11.10	自动打断块	244
11.11	有关块重定义与属性遗失	249
11.12	摘要	249
第十二章	使用 AutoLISP 参数程序系统绘制多组零件并为材料贴标签	250
12.1	参数系统能作什么	251
12.2	参数系统的组成	251
12.3	设计一个参数系统及对应屏幕菜单	251
12.4	设计外部文件格式供参数系统使用	254
12.5	提供在线说明	256
12.6	取得外部参数系统文件数据	257
12.7	产生参数系统的图形	259

12.8	绘制 90 度弯管的侧视图	261
12.9	建立多个零件(参数系统的优点)	264
12.10	绘制连接零件部份—dbline 函数	266
12.11	在管路图上加上材料注解标签	268
12.12	摘要	272
第十三章	使用 Lotus 及 Dbase 把 AutoCAD 数据库作成报表	273
13.1	建立属性抽取样板文件	274
13.2	输入属性抽取文件至 Lotus 中	277
13.3	将数据再送回 AutoCAD 图形中	290
第十四章	在 3D 空间绘图中应用 AutoLISP 的概念与工具	292
14.1	2D 与 3D 点实体的相同之处	293
14.2	是什么使 AutoCAD 具真正 3D 能力	294
14.3	检验 3D 实体的实体数据	296
14.4	在不同坐标系统之间转换 3D 点坐标	299
14.5	3D 绘图工具	302
14.6	如何经由 UCS 坐标产生多视图	302
14.7	UCS 改变引起多视区(Viewports)屏幕重绘	304
14.8	改变 UCS 平面的替代办法	305
第十五章	利用 AutoLISP 与 Scripts 进行批处理作业的程序设计	306
15.1	比较各种批处理处理方法	307
15.2	该于何时何处使用 script 或 AutoLISP 或 dxf 文件才恰当	313
15.3	建立一个批处理命令文件构造器	314
15.4	建立特殊的命令文件配置程序	321
15.5	结合 script(命令文件)与 AutoLISP 的运用	324
15.6	摘要	330
第十六章	Dxf 文件与更快速的外部文件处理的应用	331
16.1	图形转换格式	332
16.2	了解 Dxf 码及数据	336
16.3	加载 Dxf 文件	344
16.4	写个 PASCAL 程序来处理 Dxf 文件	346
16.5	摘要	348
第十七章	管理用户的系统	349
17.1	管理用户的 AutoCAD 制图系统	350
17.2	控制绘图初始设置	353
17.3	ACAD.LSP 文件	353
17.4	S::Startup—自动启动函数	355
17.5	控制命令的可用性	355
17.6	控制键盘	358
17.7	能建立 AutoLISP 宏函数命令的 Mfly.lsp 程序	361

17.8	控制 AutoLISP 的错误状况	362
17.9	建立一个错误捕捉系统	363
17.10	将错误捕捉加到程序中	367
17.11	控制函数的加载	369
17.12	管理内存	371
17.13	使用扩展 AutoLISP	374
17.14	加密码与保密	375
17.15	加密码于 LISP 程序及 menu 菜单	375
17.16	应用程序的说明与使用接口	377
17.17	LISP 程序的注解说明	377
17.18	摘要	379
17.19	作者的话	379
附录 A	供 AutoLISP 查询的 AutoCAD 命令摘要表	380
A.1	供 AutoLISP 查询的 AutoCAD 命令摘要表	380
A.2	AutoCAD R11 尺寸标注方式命令摘要表	393
A.3	AutoCAD R11 实体选取方式摘要表	394
A.4	AutoCAD R11 “点”输入格式摘要表	394
附录 B	AutoCAD R11 系统变量摘要表	396

第零章 > > AutoLISP 函数中英文 参考总表及实例

> > 以下提示的每个AutoLISP函数都附有句型、英文解释、中文解释、举例等，并附其功能及返回值的简短描述。

句型中 [XXX]代表可有可无的选择项

...代表可多添一些参数

句型:

(+ number number ...)

英文解释:

Returns the sum of all numbers, as integers or real numbers depending on the values

中文解释:

> > 依 number 为整数或实数返回所有 number 的整数或实数和。

举例:

(+ 3 2) -> 5

(+ 1 0 3 4.5) -> 8.5

(- number number ...)

Returns the difference of the first number subtracted from the sum of the remaining numbers. An integer or real number is returned, depending on the value.

> > 依number为整数或实数返回第一个number与其它number和的整数或实数差。

(- 50 30) -> 20

(- 50 40.0 2) -> 8.0

(* number number ...)

Returns the product of all numbers.

>> 返回所有 number 的乘积。

(* 2 4) -> 8

(* 2 2 4.0) -> 16.0

(/ number number ...)

Returns the quotient of the first number divided by the product of the remaining numbers.

>> 返回第一个 number 除以其它 number 乘积的商。

(/ 100 2) -> 50

(/ 100 2.0) -> 50.0

(/ 100 20.0 5) -> 1.0

(= atom atom ...)

Returns T if atoms are numerically equal, otherwise returns nil. Only numbers and strings are valid.

>> 若各atom数值相等，则返回T，否则返回nil。本函数仅对数值及字符串的比较有效。其中字符串比较是指两字符串依序比较字母的 ASCII 码值。

(= 3 3.0) -> T

(= 20 38) -> nil

(= 2.4 2.4 2.4) -> T

(= 423 423 500) -> nil

(= "mu" "mu") -> T

(= "mu" "you") -> nil

(/= atom atom)

Returns T if atoms are numerically not equal, otherwise returns nil. Only numbers and strings are valid.

>> /= 代表不相等，若各atom数值不相等，返回T，否则返回nil。本函数仅对数值及字符串的比较有效。

(/= 12 20) -> T

```
(/ = "your" "your")    -> nil  
(/ = 5.41 5.47)        -> T
```

(< atom atom ...)

Returns T if each atom is numerically less than the following atom, otherwise returns nil. Only numbers and strings are valid.

>> 若各左方atom数值均小于其右方的atom, 返回T, 否则返回nil。本函数仅对数值及字符串的比较有效。

```
(< 12 22)              -> T  
(< "a" "c")            -> T  
(< 38 33.2)            -> nil  
(< 2 3 4 4)            -> nil
```

(<= atom atom ...)

Returns T if each atom is numerically less than or equal to the following atom, otherwise returns nil. Only numbers and strings are valid.

>> 若各左方atom数值小于等于其右方的atom, 返回T, 否则返回nil。本函数仅对数值及字符串的比较有效。

```
(<= 12 20)             -> T  
(<= "d" "d")           -> T  
(<= 38 33.2)           -> nil  
(<= 3 8 8)              -> T  
(<= 2 8 4 5)           -> nil
```

(> atom atom ...)

Returns T if each atom is numerically greater than the following atom, otherwise returns nil. Only numbers and strings are valid.

>> 若各左方atom数值大于其右方的atom, 返回T, 否则返回nil。本函数仅对数值及字符串的比较有效。

```
(> 122 18)             -> T  
(> "d" "b")            -> T  
(> 3.4 179)            -> nil  
(> 78 5 2)             -> T
```

(> 69 4 4) -> nil

(> = atom atom ...)

Returns T if each atom is numerically greater than or equal to the following atom, otherwise returns nil. Only numbers and strings are valid.

>> 若各左方atom数值上大于等于其右方的atom, 返回T, 否则返回nil. 本函数仅对数值及字符串的比较有效.

(> = 122 18) -> T

(> = "g" "g") -> T

(> = 3.5 178) -> nil

(> = 78 4 4) -> T

(> = 78 5 8) -> nil

(~ number)

Returns the bitwise NOT of number.

>> 将number数各个位值0改为1, 1改为0后, 返回其值.

(~ 100) -> -101

(~ -4) -> 3

(1+ number)

Returns number incremented by 1.

>> 返回 number 加 1 后的值.

(1+ 8) -> 9

(1+ -27.3) -> -26.3

(1- number)

Returns number decremented by 1.

>> 返回 number 减 1 后的值.

(1- 8) -> 7

(1- -15.4) -> -16.4

(abs number) **abs** 表示 absolute value

Returns the absolute value of an integer or real number.

> > 返回 number 整数或实数的绝对值。

(abs -80) -> 80

(abs -92.43) -> 92.43

(and expression ...)

Returns T if all expressions are true, otherwise returns nil and ceases evaluation at the first nil expression encountered.

> > expression 可以是常数、变量、字符串等各种 AutoLISP 的数据，或是一个数学表达式，若所有 expression 皆成立，不为 nil，返回 T，否则返回 nil；在遇到第一个不成立的 expression 之后就返回 nil，不必再计算其它 expression 了。

假设 (setq a 103)

 (setq b nil)

 (setq c "string")

则

(and 1.4 a c) -> T

(and 1.4 a b c) -> nil

(angle point point)

Returns angle in radians from the X axis in a counterclockwise direction to a line between the two points.

> > 返回由 X 轴依逆时针方向转到两个 point 参数连线方向的夹角，单位是弧度。

(angle '(1.0 1.0) '(1.0 4.0)) -> 1.5708 (单位为radian弧度)

(angle '(5.0 1.33) '(2.4 1.33)) -> 3.14159 (单位为radian弧度)

(angtos angle [mode] [precision]) **angtos** 表示 angle to string

Returns a string conversion of an angle from radians to the units specified by mode. The conversion defaults to the current angular units mode and precision unless otherwise specified with the optional mode and precision arguments.

> > 将弧度单位的 angle 参数角度值以 mode 模式及 precision 精度转换成字符串。除非重新赋值 mode 及 precision 参数值。否则 mode 值参考 AUNITS 系统变量，

precision 值参考 AUPREC 系统变量。angtos 接受负值的输入，转换成 0 至 $2\pi(\pi = 3.14)$ 之间的正值返回。

假设(setq pt1 '(5.0 1.33))

(setq pt2 '(2.4 1.33))

(setq a (angle pt1 pt2))

则

(angtos a 0 0)	-> "180"	
(angtos a 0 4)	-> "180.0000"	<-十进制格式
(angtos a 1 4)	-> "180d0'0"	<-度分秒格式
(angtos a 3 4)	-> "3.1416r"	<-弧度格式
(angtos a 4 2)	-> "W"	<-“西方”方位格式
(angtos 0.785398 0 4)	-> "45.0000"	
(angtos -0.785398 0 4)	-> "315.0000"	

(append list)

Returns a single list made up of any number of lists.

> > 返回，由各 list 参数组成的单一表。

(append '(a b) '(c d)) -> (A B C D)

(append '((a) (b)) '((c) (d))) -> ((A)(B)(C)(D))

(apply function list)

Applies a function to the arguments supplied by list. Generally, function and list are quoted so their contents will be supplied unevaluated.

> > 将function函数作用于list所提供的诸参数上。通常function及list皆以引号开头，提供其不被计算的内容。

(apply '+ '(1 2 6)) -> 9

(apply 'strcat '("a" "b" "c")) -> "abc"

(ascii string)

Returns the first character of the string as an ASCII integer character code.

> > 返回 string 字符串开头字母对应 ASCII 整数码

(ascii "A") -> 65

(ascii "a") -> 97

(assoc item list) assoc 表示 associate

Returns a list containing the item from a list of lists.

>> 返回 list 参数(包含多个子 list)中, 包含 item 参数的子 list

假设(setq al '((name box) (width 3) (size 4.7263) (depth 5)))

则

(assoc 'size al) -> (size 4.7263)

(assoc 'weight al) -> nil

(atan number1 [number2]) atan 表示 arctangent

Returns the arctangent of number1, from $-\pi$ to π . If number2 is provided, the arctangent of the number1 / number2 is returned. If number2 is 0 either $-\pi/2$ or $+\pi/2$ radians (-90 or $+90$ degrees) is returned, depending on the sign of number1.

>> 返回第一个 number1 的反正切函数值, 其值范围在 $-\pi$ 到 π . 若提供第二个 number2 的参数, 则返回 number1 / number2 的反正切函数值. 若 number2 为 0, 依 number1 的正负号返回 $-\pi/2$ 或 $+\pi/2$

(atan 0.5) -> 1.463648

(atan 1.0) -> 0.785398

(atan -1.0) -> -0.785398

(angtos (atan -1.0) 0 4) -> "315.0000"

(atof string) atof 表示 ascii to floating

Returns a real number converted from a string.

>> 将 string 字符串转换为实数返回.

(atof "92.8") -> 92.8

(atof "4") -> 4.0

(atoi string) atoi 表示 ascii to integer

Returns an integer converted from a string.

>> 将 string 字符串转换为实数返回.

```
(atoi "95")      -> 95
(atoi "2.9")     -> 2
```

(atom item)

Returns T if item is not a list, otherwise returns nil.

>> 当 item 参数不为表时就算是个 atom, 返回 T, 否则返回 nil.

假设(setq a '(x y z))

(setq b 'a)

则

```
(atom 'a)          -> T
(atom a)           -> nil
(atom 'b)          -> T
(atom b)           -> T
(atom '(a b c))    -> nil
```

(boole function integer integer ...) boole 表示 boolean

Returns one of 16 possible boolean operations, based on the function value, on any number of integers.

>> 返回16种布尔运算之一的结果值。根据function值及诸integer值来决定作那一种布尔运算。

```
(boole 1 12 5)      -> 4 (即12和5的and运算)
(boole 6 6 5)       -> 3 (即6和5的Xor运算)
```

(boundp atom) boundp 表示 bound property

Returns T if the atom is bound to a value, otherwise returns nil

>> 当 atom 有设置值(赋予某值)时, 返回 T, 否则返回 nil.

假设(setq a 2)

(setq b nil)

则

```
(boundp 'a) -> T
(boundp 'b) -> nil
```

(cadr list)

Returns the second element in the list. Use CADR to extract the Y coordinate of a pointlist.

> > 返回 list 的中第二个元素，通常使用 cadr 函数抽取点坐标序列的 Y 坐标。

(car list)

Returns the first element in the list. Use CAR to extract the X coordinate of a pointlist.

> > 返回 list 中的第一个元素，通常使用 car 函数抽取点坐标序列的 X 坐标。

```
(car '(a b c))          -> A
(car '((a b) c))        -> (A B)
(car '())                -> nil
```

(cdr list)

Returns a list minus the first element in the list.

> > 返回表中减去第一个元素后剩下的表。

```
(cdr '(a b c))          -> (B C)
(cdr '((a b) c))        -> (c)
(cdr '())                -> nil
(cdr '(a .b))            -> B
(cdr '(1 . "Text"))      -> "Text"
```

(c????r list)

Returns an element or list defined by the combinations of a and d characters in the expression, up to four levels deep. For example, caadr, caddr, cadar, and so on.

> > 此式中的????代表a与d的结合，至多四个a或d，例如caadr, caddr, cadar等。????执行时是由右往左，d代表删除其第一个元素后剩下的表；a代表表中的第一个元素，最后的返回值可能是个元素或表。

假设(setq x '((a b) c d))

```
(caar list) = (car (car list))          -> A
```