

Remote Sensing of Secondary
Development Language IDL

遥感二次开发 语言 IDL

徐永明 编著 ©



科学出版社

遥感二次开发语言 IDL

徐永明 编著

科学出版社

北京

内 容 简 介

本书的主旨是让读者掌握 IDL 语言编程方法,并灵活应用到具体的遥感问题中去。书中以 IDL8.2 为基础,系统介绍 IDL 编程语言及其在遥感相关领域的实际应用。全书共 9 章,涵盖 IDL 语言概述、IDL 语法基础、IDL 编程基础、数据的读写操作、图形绘制、图像处理、随机数、统计与插值、IDL 与 ENVI 结合等方面的内容,并提供大量的 IDL 遥感应用实例。

本书力求通俗易懂,简洁明了,注重理论联系实际,可作为高等院校、科研院所的遥感及相关专业本科生及研究生的参考教材及自学用书,也可供从事相关领域工作的广大科研人员和技术人员学习和参考。

图书在版编目(CIP)数据

遥感二次开发语言 IDL / 徐永明编著. —北京:科学出版社,2014.6

ISBN 978-7-03-041185-3

I. ①遥… II. ①徐… III. ①遥感技术—软件工具—程序设计

IV. ①TP7②TP311.56

中国版本图书馆 CIP 数据核字(2014)第 129035 号

责任编辑:伍宏发 刘燕春/ 责任校对:钟 洋

责任印制:肖 兴/ 封面设计:许 瑞

科学出版社出版

北京东黄城根北街 16 号

邮政编码:100717

<http://www.sciencep.com>

骏杰印刷厂印刷

科学出版社发行 各地新华书店经销

*

2014 年 6 月第 一 版 开本:787×1092 1/16

2014 年 6 月第一次印刷 印张:18 1/4

字数:432 000

定价:68.00 元

(如有印装质量问题,我社负责调换)



前 言

卫星遥感是一门新兴的空间信息科学，能够以宏观、动态、快捷的手段获取空间上连续的地表及近地表信息，在环境监测、资源调查、土地管理、城市规划、矿产勘探、公共健康、国防安全、农林管理、气象气候等多个领域发挥着越来越重要的作用，并对社会经济和科学研究具有重要意义。IDL 语言作为一门面向矩阵的编程语言，在遥感图像处理和信息提取方面有着独到的优势，其与遥感通用处理软件 ENVI 的结合，更是为遥感方面的工作提供了强有力的工具。

笔者自 2004 年开始在遥感科研工作中接触 IDL 语言，并从 2008 年开始承担遥感科学与技术专业本科生《遥感二次开发语言》的教学工作，由于没有适合该课程的 IDL 教材，因此自行编写了课程讲义，并根据教学和科研工作中的经验，不断进行修改完善和补充，最终完成本书。

本书是为遥感科学与技术专业本科生的遥感二次开发语言课程所编写的专业教材，同时也可供相关专业的本科生、研究生及相关领域的科技工作者学习和参考。为了适应遥感学科教学的实际需要，本书对 IDL 语法部分进行适当取舍，并提供了大量翔实的应用实例，知识容量适度，理论联系实际。全书内容共分 9 章：第 1 章主要对 IDL 语言进行简单介绍；第 2 章主要介绍 IDL 的基础语法，包括变量、数组；第 3 章主要介绍 IDL 的编程基础，包括过程/函数、控制语句、参数/关键字及变量作用域等；第 4 章主要介绍数据的读写操作，包括文件基本操作、ASCII 码文件读写、二进制文件读写、图像文件读写以及 HDF、netCDF 等文件内容的读取等；第 5 章主要介绍 IDL 图形绘制方法，包括曲线图、散点图、柱状图及直方图；第 6 章主要介绍 IDL 图像处理，包括图像的显示、统计、增强、滤波及几何变换等；第 7 章主要介绍 IDL 随机数、统计与插值等；第 8 章主要介绍 IDL 与 ENVI 的结合，包括两者间的数据交互、ENVI 调用 IDL 函数以及 IDL 调用 ENVI 函数等；第 9 章则通过 20 个具体的遥感实例展示 IDL 编程语言在遥感数据处理及应用中的作用。

本书在南京信息工程大学教材基金项目“遥感二次开发语言 IDL”的资助下出版，在此表示诚挚的谢意！

由于笔者水平的限制，书中错误与疏漏之处在所难免，恳请广大读者批评指正。

徐永明

2014 年 4 月

目 录

前言

第 1 章 IDL 语言概述	1
1.1 IDL 语言简介	1
1.2 IDL 工作环境	1
1.2.1 IDL 工作界面	1
1.2.2 IDL 帮助	2
1.3 IDL 使用时的一些注意事项	4
1.3.1 本书中 IDL 代码的表达方式	4
1.3.2 IDL 注意事项	4
1.4 本书的说明	5
第 2 章 IDL 语法基础	6
2.1 变量	6
2.1.1 变量概述	6
2.1.2 数据类型	6
2.1.3 变量基本操作	7
2.2 数组	10
2.2.1 创建数组	10
2.2.2 数组的下标	13
2.2.3 数组操作函数	14
2.3 字符串	21
2.3.1 创建字符串	21
2.3.2 字符串连接	21
2.3.3 字符串操作函数	22
2.3.4 字符串与数值的相互转换	26
2.3.5 字符串读取	27
2.4 表达式	28
2.4.1 数值型表达式	28
2.4.2 字符型表达式	29
2.4.3 关系型表达式	29
2.4.4 逻辑型表达式	30
2.4.5 条件表达式	31
2.4.6 位运算	31
2.4.7 数组运算	33

2.4.8	运算符的优先级	34
2.4.9	常用的数学运算函数	35
2.5	时间	35
2.5.1	系统时间	35
2.5.2	时间格式转换	36
2.5.3	其他的时间操作	38
2.6	结构体	40
2.6.1	匿名结构体	40
2.6.2	署名结构体	41
2.6.3	结构体数组	42
2.6.4	结构体操作函数	42
2.7	指针	44
2.7.1	指针的创建	44
2.7.2	指针的提取	44
2.7.3	指针的释放	45
2.7.4	指针的验证	45
2.7.5	指针数组	45
第 3 章	IDL 编程基础	46
3.1	过程和函数	46
3.1.1	过程	46
3.1.2	函数	47
3.1.3	程序的相互调用	48
3.2	控制语句	48
3.2.1	选择结构	49
3.2.2	循环结构	53
3.2.3	continue 和 break 语句	55
3.3	参数和关键字	57
3.3.1	参数	57
3.3.2	关键字	58
3.3.3	值传递和地址传递	59
3.3.4	参数和关键字的检测	59
3.4	变量的作用域	60
3.4.1	局部变量	60
3.4.2	全局变量	61
3.5	其他	61
3.5.1	IDL 程序优化	61
3.5.2	调用外部命令	64

第 4 章 数据的读写操作	65
4.1 标准输入输出	65
4.1.1 标准输出	65
4.1.2 标准输入	68
4.2 文件的相关操作	69
4.2.1 文件的打开与关闭	69
4.2.2 文件的其他操作	70
4.3 读写 ASCII 码文件	75
4.3.1 读取 ASCII 码文件	75
4.3.2 写入 ASCII 码文件	81
4.3.3 读写 CSV 文件	82
4.4 读写二进制文件	83
4.4.1 读取二进制文件	83
4.4.2 写入二进制文件	85
4.5 读写图像文件	86
4.5.1 图像文件查询	86
4.5.2 读取图像文件	88
4.5.3 写入图像文件	89
4.6 读取 HDF 文件	90
4.6.1 读取 HDF4 文件	91
4.6.2 读取 HDF5 文件	93
4.7 读取 netCDF 文件	95
第 5 章 图形绘制	98
5.1 plot 过程绘曲线图	98
5.1.1 基本曲线图	98
5.1.2 线型设置	99
5.1.3 坐标轴设置	101
5.1.4 颜色设置	102
5.1.5 添加标注	104
5.1.6 绘制多幅图形	108
5.1.7 图形保存为文件	109
5.2 plot 函数绘曲线图	111
5.2.1 基本曲线图	111
5.2.2 线型设置	112
5.2.3 符号设置	114
5.2.4 坐标轴设置	115
5.2.5 绘制多幅图形	116
5.2.6 图形对象操作方法	116

5.2.7	添加标注	118
5.2.8	添加图例	120
5.3	柱状图	123
5.4	直方图	126
第 6 章	图像处理	128
6.1	图像显示	128
6.2	图像统计	131
6.2.1	常规统计	131
6.2.2	直方图统计	132
6.3	图像增强	134
6.3.1	线性增强	134
6.3.2	直方图均衡	136
6.3.3	掩膜运算	137
6.3.4	密度分割	138
6.3.5	色彩空间变换	142
6.4	图像滤波	143
6.4.1	平滑滤波	143
6.4.2	锐化滤波	145
6.4.3	卷积运算	146
6.5	图像几何变换	146
6.5.1	图像裁切	146
6.5.2	图像重采样	146
6.5.3	图像转置	147
6.5.4	图像旋转与翻转	148
第 7 章	随机数、统计与插值	150
7.1	随机数	150
7.1.1	生成随机数	150
7.1.2	随机数的应用	151
7.2	相关分析	153
7.3	回归分析	155
7.3.1	线性回归	155
7.3.2	显著水平检验	158
7.3.3	曲线拟合	159
7.4	插值	164
第 8 章	IDL 与 ENVI 的结合	169
8.1	IDL 与 ENVI 的数据交互	169
8.2	ENVI 调用 IDL 函数	170
8.2.1	波段运算函数	170

8.2.2 波谱运算函数	171
8.3 IDL 调用 ENVI 功能	172
8.3.1 常用的 ENVI 函数	173
8.3.2 envi_doit 处理函数	195
8.3.3 影像分块	209
第 9 章 IDL 遥感应用实例	211
9.1 光谱数据处理	211
9.2 高光谱数据最佳波段组合选择	214
9.3 遥感图像信噪比计算	216
9.4 批量生成遥感影像快视图	219
9.5 辐射定标与大气校正	220
9.6 土地覆盖图空间重采样	224
9.7 MODIS L1B 数据的几何重定位处理	226
9.8 批量遥感数据的镶嵌、裁切和掩膜处理	228
9.9 NDVI 数据的 MVC 合成处理	235
9.10 ROI 数据的随机分割	236
9.11 根据站点坐标提取高程值	239
9.12 黑体辐射出射度计算	240
9.13 水体动态变化遥感监测	242
9.14 叶面积指数遥感估算	244
9.15 植被覆盖度遥感监测	246
9.16 地表温度遥感反演	249
9.17 温度植被干旱指数计算	254
9.18 气溶胶光学厚度的遥感反演	259
9.19 大气水汽总含量遥感估算	268
9.20 林火遥感监测	270
参考文献	278

第 1 章 IDL 语言概述

1.1 IDL 语言简介

交互式数据语言 IDL (Interactive Data Language) 是美国 Exelis VIS 公司开发的一种面向矩阵的计算机语言。自 1977 年问世以来, IDL 以其简单灵活的语法、强大的数据分析处理和可视化功能, 在遥感、天文、地质、航天、医学和军事等许多领域得到了广泛应用。

IDL 语言最大的特点是面向矩阵, 对于数组元素的操作可以不通过循环而直接对数组进行运算。与 C、Basic 等语言相比, IDL 对数组的操作快捷且方便。与另一种面向矩阵的语言——Matlab 相比, IDL 更适用于图像处理, 尤其是大数据量图像处理方面。因为 Matlab 的设计初衷是基于较小的二维矩阵, 强制进行双精度计算使其在计算时消耗的内存有时达到实际数据精度的 8 倍, 在造成资源浪费的同时, 束缚了处理数据量的能力。IDL 语言自动支持多线程 CPU 的计算, 并且无需在代码中体现, 从而有效提高计算速度, 降低编程难度。此外, 基于 IDL 语言开发的遥感图像处理平台 ENVI 与 IDL 语言能够方便地进行数据交互和函数调用, 大大提高了 IDL 语言处理遥感图像的能力, 可以使用户更专注于算法方面的工作, 用最少的时间和资源完成任务。因此, IDL 语言是进行遥感图像处理及应用分析的理想工具。

1.2 IDL 工作环境

1.2.1 IDL 工作界面

IDL8.2 的工作界面由菜单栏、工具栏、项目管理/变量跟踪窗口、程序编辑窗口、IDL 控制台、状态栏等构成 (图 1.1)。

菜单栏主要包括文件、编辑、源码、项目、运行、窗口和帮助等 7 个菜单项, 提供了文件管理、编辑、代码格式、项目管理、运行调试、窗口视图管理和帮助等方面的功能。

工具栏提供了常用功能的按钮, 包括打开程序文件、新建文件或工程、保存程序文件、剪切、复制、粘贴、撤销、后退、前进、编译、运行、调试和重置等按钮。用户通过单击工具栏按钮, 在编写程序时可方便快捷的实现相应操作, 其使用频率要远高于菜单栏。

项目管理/变量跟踪窗口以标签页的方式切换项目管理和变量跟踪功能。项目管理能够实现编辑、添加、删除和编译文件以及生成和运行项目文件的功能, 方便用户对项目文件进行集中管理; 变量跟踪则用于显示内存中各个 IDL 变量的名称、值、类型等。

程序编辑窗口是一个显示、输入和编辑 IDL 代码的文本编辑器, 在编写 IDL 过程或者函数时只能通过程序编辑窗口完成。

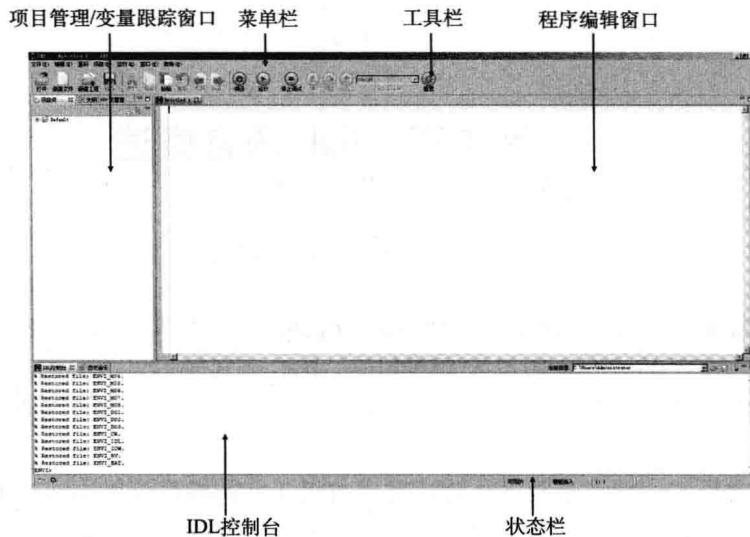


图 1.1 IDL8.2 的工作界面

IDL 控制台用于输入 IDL 命令行、显示程序执行结果。在控制台中可以输入各种 IDL 命令，命令的执行结果也在控制台中显示。IDL 能自动记忆用户输入的命令，用户可以通过“↑”、“↓”方向键或者切换到历史命令标签下查看前/后面所输入的命令，以便重复选取并执行某条命令而无需重新输入。

状态栏的主要作用是显示当前光标所在位置，即第几行第几列。此外还提供了快速启动视图、显示当前文件的基本属性等功能。

1.2.2 IDL 帮助

IDL 的帮助系统比较完善，大致可分为两类：联机帮助系统和 Demo 演示系统。用户在学习 IDL 编程的过程中，掌握和运用这些帮助系统非常重要。下面分别介绍这两类帮助系统。

1. 联机帮助系统

IDL8.2 的联机帮助系统可以通过“帮助”菜单、点击 F1 键或者在控制台中输入“?”打开，系统界面如图 1.2 所示。帮助系统包括内容 (Contents)、索引 (Index)、搜索 (Search) 和收藏夹 (Favorites) 4 种帮助方式：“内容”以目录形式分层次列出了所有帮助内容；“索引”按照字母顺序列出 IDL 所有的过程、函数、运算符、系统变量等；“搜索”可以根据用户输入的内容查询出所有符合条件的帮助信息；“收藏夹”与 IE 等网络浏览器的收藏夹功能类似，可以将用户感兴趣的帮助页面加入收藏夹，方便用户查找。

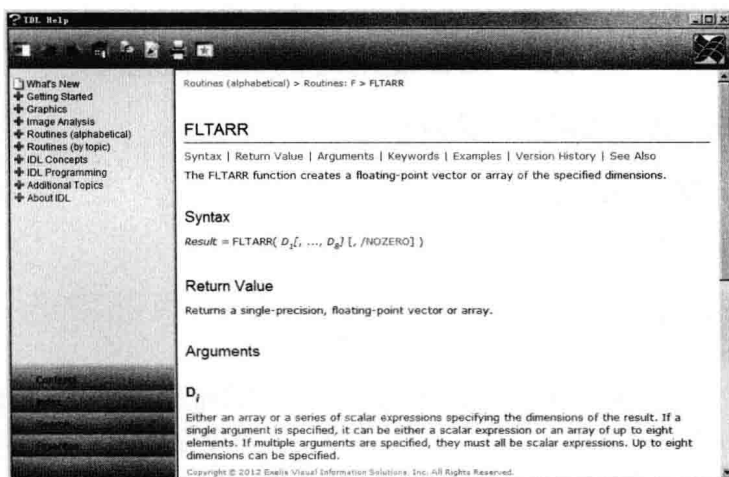


图 1.2 IDL8.2 的联机帮助系统界面

2. Demo 演示系统

Demo 演示系统可以通过在控制台中输入“demo”打开，系统界面见图 1.3。Demo 演示系统提供了官方的一些数据分析以及可视化方面的例子，包括数学统计、信号处理、插值计算、图像处理、矢量绘图、三维显示、三维体数据显示等。用户在 Demo 系统界面左侧的目录树中选择具体分类，右侧则会显示该分类下具体的 Demo 例子，双击可打开该 Demo。此外，Demo 对应 IDL 代码的路径和文件名会出现在下面的状态栏中，用户可按照该路径和文件名去查看该 Demo 的 IDL 源代码。

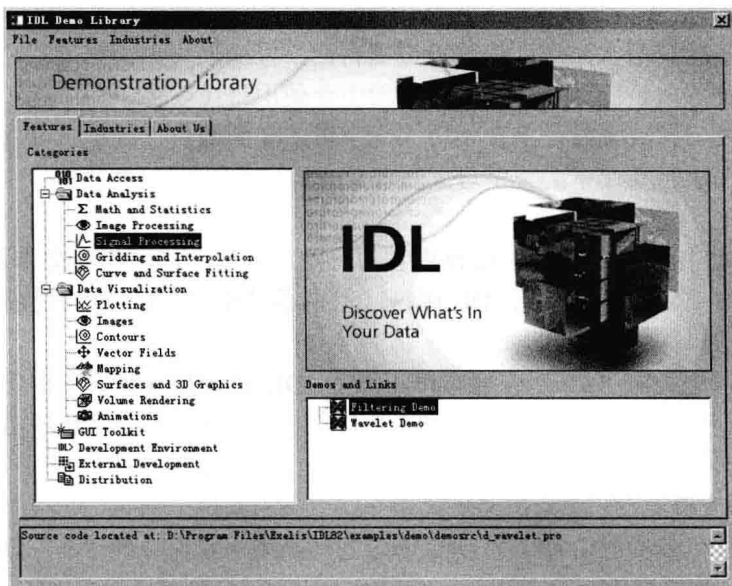


图 1.3 IDL8.2 的 Demo 演示系统界面

1.3 IDL 使用时的一些注意事项

1.3.1 本书中 IDL 代码的表达方式

本书中，IDL 代码以下面两种方式表达：

1. 命令行

```
IDL> print, "Hello world"  
Hello world
```

“IDL>”开头表示在 IDL 控制台中以命令行方式输入（如果是“ENVI>”开头则表示在 ENVI+IDL 工作模式下的命令行方式输入），命令行下面没有用“IDL>”开头则为 IDL 语句的输出结果。

2. 过程/函数

```
pro temp  
    print, 'Hello world'  
end
```

以 `pro` 或者 `function` 开头、以 `end` 结尾的语句表示 IDL 过程或者函数，一般在 IDL 界面的程序编辑窗口输入。

1.3.2 IDL 注意事项

在编程 IDL 程序过程中，以下事项需要注意：

1) 大小写

IDL 语言不区分大小写，用户可根据自己喜好来采用大小写方式。相对而言，小写字母比较方便输入，因而在编写代码过程中会使用得比较多。为了便于阅读程序，也可以根据自己的习惯或喜好，部分使用大写字母，如函数名、过程名及关键字等用大写或者首字母大写、重要变量的首字母大写等。

2) 注释

IDL 语言中分号“;”表示注释内容的开始，分号右边的任何文本都被视为是注释，在程序执行时将被忽略。在写 IDL 代码过程中适量使用注释，可以提高程序的可读性。

3) 续行符

IDL 语言中符号“\$”用作续行符，表示 IDL 语句延续到下一语句行。如果某条语句过长，超出程序编辑窗口宽度，不利于程序阅读和修改，可以将这条语句分成若干段，并用续行符连接起来。

4) 续命令符

一行可以写多条 IDL 语句，但是语句之间必须用续命令符“&”分隔，表示这是多条语句，IDL 将分别执行这些语句。

1.4 本书的说明

本书通过 IDL 基本语法及大量的实例介绍了 IDL 语言在遥感中的应用。目的在于帮助读者在遥感工作过程中遇到现有遥感软件无法解决或者不便解决的任务时，能够通过 IDL 语言编程解决问题。侧重点是以尽可能简单的编程语句实现目标，而不是开发具有完善图形界面的遥感处理平台。简而言之，本书意欲帮助读者编写出“自己用的”IDL 程序，而不是写出给“别人用的”程序。因此，本书主要讲述的是 IDL 的基本语法以及与遥感密切相关的一些内容（包括矢量绘图、图像处理、统计分析、ENVI 二次开发、遥感编程实例等），并未涉及对象图形、界面及事件处理、数据库等内容。

本书的所有代码均在 IDL8.2 下测试通过，除了特别指出在新版本中才增加的功能外，其余代码也可以在 IDL6.3 下运行。

第 2 章 IDL 语法基础

2.1 变 量

2.1.1 变量概述

变量是指在程序运行过程中其值可以发生变化的数据。IDL 变量的命名规则如下：

- (1) 第一个字符必须为英文字母；
- (2) 必须由英文字母、数字、下划线和美元符号“\$”组成；
- (3) 长度不超过 128 个字符；
- (4) 中间不能有空格；
- (5) 变量名不能是系统内部用于特殊用途的保留字名称（IDL 保留字见表 2.1）。

表 2.1 IDL 保留字

AND	BEGIN	CASE	COMMON	DO	ELSE	END
ENDCASE	ENDELSE	ENDFOR	ENDIF	ENDREP	EQ	FOR
FUNCTION	GE	GOTO	GT	IF	LE	LT
MOD	NE	NOT	OF	ON_IOERROR	OR	PRO
REPEAT	THEN	UNTIL	WHILE	XOR		

除了用户自己定义的变量以外，IDL 还提供了一些已经定义好的特殊变量，这些变量以感叹号“!”开头，称为系统变量。系统变量的类型和结构不能改变，也不能从内存中释放。表 2.2 给出了几个常用的系统变量，所有的系统变量信息可参考 IDL 帮助。

表 2.2 IDL 常用的几个常数系统变量

变量名	说明
!PI	圆周率 π （单精度），3.141 59
!DPI	圆周率 π （双精度），3.141 592 7
!DTOR	角度到弧度的转换系数， $\pi/180$
!RADEG	弧度到角度的转换系数， $180/\pi$
!COLOR	系统色彩查找表，可根据色彩名称获取 RGB 值

2.1.2 数据类型

IDL 的数据类型可以分为两类：11 种数字数据类型（表 2.3）和 6 种非数字数据类

型（表 2.4）。此外，任何 IDL 数字类型或者非数字类型变量都可以作为一维或多维数组的元素。

表 2.3 IDL 数字数据类型

数据类型	描述	字节数	范围
byte	字节型	1	0~255
int	整型	2	-32 768~32 767
uint	无符号整型	2	0~65 535
long	长整型	4	$-2^{31} \sim 2^{31} - 1$
ulong	无符号长整型	4	$0 \sim 2^{32} - 1$
long64	64 位长整型	8	$-2^{63} \sim 2^{63} - 1$
ulong64	64 位无符号长整型	8	$0 \sim 2^{64} - 1$
float	浮点型	4	$-10^{38} \sim 10^{38}$
double	双精度浮点型	8	$-10^{308} \sim 10^{308}$
complex	复数	8	$-10^{38} \sim 10^{38}$
dcomplex	双精度复数	16	$-10^{308} \sim 10^{308}$

表 2.4 IDL 非数字数据类型

数据类型	描述
string	字符串（0~32 767 个字符）
struct	结构体，一个或者多个变量的组合
pointer	指针
object	对象
list	链表
hash	哈希表

2.1.3 变量基本操作

IDL 变量在使用前不需要事先声明，也不需要指定类型，可通过赋值形式直接进行定义，并且随时可以改变数据类型和维数。

```
IDL> a=1.1
IDL> help, a
A          FLOAT    =    1.10000
IDL> a=2B
IDL> help, a
A          BYTE     =    2
```

通过赋值语句直接将变量 a 定义为一个浮点型变量，值为 1.1，然后又通过另一条赋

值语句转变为值为 2 的字节型变量。创建各种变量的方法详见表 2.5。

表 2.5 IDL 变量的创建与转换

数据类型	字节数	创建变量	数据类型转换函数
字节型	1	0b	byte()
整型	2	0	fix()
长整型	4	0l	long()
64 位长整型	8	0ll	long64()
无符号整型	2	0u	uint()
无符号长整型	4	0ul	ulong()
64 位无符号长整型	8	0ull	ulong64()
浮点型	4	0.0	float()
双精度浮点型	8	0.0d	double()
复数	8	complex(0.0,0.0)	complex()
双精度复数	16	dcomplex(0.0d,0.0d)	dcomplex()
字符串	0~32 767	''或'''	string()
指针	4	ptr_new()	none()
对象	4	obj_new()	none()

在上面的代码中用到了 `help` 命令，该命令可以帮助用户跟踪变量类型和大小。对于标量变量，`help` 命令显示变量的名称、类型和值。对于数组型变量，`help` 命令显示变量的名称、类型和大小。

```
IDL> value=1.23
IDL> help, value
VALUE          FLOAT      =      1.23000
IDL> arr=[1, 2, 3, 4]
IDL> help, arr
ARR            INT        = Array[4]
```

`print` 命令可以直接显示标量和数组变量的值。

```
IDL> arr=[1.1, 2.6, -1.1, -2.6]
IDL> print, arr
      1.10000      2.60000      -1.10000      -2.60000
```

需要特别说明的是，在科学研究中时常会遇到特别大的数值，比如普朗克定律中的普朗克常数 ($6.63 \times 10^{-34} \text{ J}\cdot\text{s}$)、光速 ($2.998 \times 10^8 \text{ m}\cdot\text{s}^{-1}$)、波尔兹曼常数 ($1.38 \times 10^{-23} \text{ J}\cdot\text{K}^{-1}$)。这些数值一般需要通过科学计数法来表达，而 IDL 提供了这样的赋值方法：

```
IDL> a=6.63e-34
IDL> help, a
A          FLOAT      = 6.63000e-034
```