

IDL 逐步: 定位参数, 关键字参数 ~ 可以不用整写全种单词.

过程及函数: $x_{tickvalue} \rightarrow x_{tickv}$

变量: 4种基本类型: P13
Long, Long64, Float, Double
(int4) (int8) (real4) (real8)
OL OLL 0.0 0.0D

解月日: systime / Julian
结构体: typedef struct {

数组 & 数组: [[1, 2, 3] [3, 2, 1]] 2行 3列
万括号, 0开始
reform (vec, 2, 3): 2行 3行

初始化 (为0):
xxArr: LonArr, FltArr, ...
P18 索引值: LIndgen, FIndgen, ...

* 表示该维度所有值 -> 在 Fortran 中为: "

随机数: RandomU (seed, 10, 20), 列行

合并 [a, b] (a) (b) -> (c)
[[a], [b]] {a} {b} -> (c)

常量: !pi, !dpi, ...

运算: 求最大 >, 求最小 <, 取模 mod.

矩阵乘法: A1 # A2 = A2 A1
A1 # # A2 = A1 A2

字符: + <-> Fortran 中为 "

关系: EQ, NE, GE, GT, LE, LT (无点)

逻辑: &&, ||, -

库函数: 取整: ABS, EXP, ceil, floor, round, Avg, Avglo, Sqrt

字符: SystemE (utc) 或 Julian
stramp (A1, A2, [N], [Fold-case])
strampress (按空) 如上, range-all ~ 全去.

数组: Max, Min, N-elements, Size, Mean, Variance, Stdev
元素 形状 dim 维数 标准差

WHERE, REVERSE, rotate, reform, transpose, shift, sort
向下转 逆 转 转置 转置 排序
rebin, Congrid (P38) -> 调整大小

结构化: For ... DO ... / For ... DO begin
Endfor > 先判断后执行
WHILE (... DO ... / WHILE ... DO begin
Endwhile
Repeat ... Until ... / Repeat begin (先执行后判断)
Endrep until ...

选: if ... then ... (else -) / if ... then begin
endif [else begin ...] endif
elseif ... endif
endelse
Switch x of
1: ...
2: ...
...
else ...
endswitch
以后所有均执行 (仅执行一句)

读写数据: open r/w/u (读/写) OPENR, LUN, ... / Get_Lun
close, LUN & FREE_LUN, LUN
判断文件 & 逻辑设备号相连: Help, / Files
自由: Print (copenw) ~ P60: 读: ReadF. (读 P3-4)
Bjpeg.gif, P104-115 (读/写)
Fortran 数据: OPENR, LUN, ... / Get_Lun, / FT - Unformatted
DATA = (FArr (2,4) & READU: LUN, DATA)

图形显示:
plot 语句: title, xtitle, ytitle / isotropic -> 正交的
charsize, xcharsize, ycharsize (还有 !P: charsize
字型 -> 粗细: 符号 (PSym) -> SymSize ~ 还可自创 (P127).
P124 thick P125
不同颜色绘图 (Color, Background) 框号或: Color
颜色: / nodata & Dplot
范围: X/Y range; 风格: X/Y style; 网格线 X/Y tick len = 1
XY Grid style = 线型.
刻度: X/Y ticks, X/Y minor.
(主要) 多数据: Dplot 或 Plots (并智能)
共用 (右侧) X轴: Axis, [Axis = 1, [range = [J, / save ...
~ tick 两行: P138.

误差棒: errplot P140 -> 双向:
表面图: Surface, Az = ..., Ax = ... (绕 Z 轴旋转一度, 绕 X 轴旋转一度).
颜色: (P148), 外观 style (常用) P140
-> Shade_Surf, Light, ... (Set - Shading).

等值线图: Contour 语句.
Nlevels = 12 (12等分) / levels = 10 (Findgen(5) - 4)
Clabels = [...] / Replicate (1, Nlevels) -> C-annotation (...)
均作图 均作图 -> C-Linestyle, C-thick, C-downhill, C-colors
fill / overlay / P170
(levels = 20) (Nlevels = 2)

定位图形: !P: position, !P: Region / margin & origin, margin
位置 边缘

多图: / noerase
!P: multi 明确指定 2 图: [0, 2, 3, 0, 1]
构造: X (units) / 最上端 + title = ...
不对称: 临时设 !P: multi = [1, ...]

添加文本: 零件 !P: Font = 1.0, 1 (P25) 字体大小 P187 字体条件
P187, P192.
X/Y axis, x, y, s, size, ... / normal.
Alignment = 0 左 / 1 右 (按列对齐); 轴上文本
Orientation: 正 (竖 / 横).

画线及符号: dplot 或 Plots 更智能; 颜色: Polyfill (Bool)
着色: Device, Decomposed = 0: win.
Loadit, 32L 或 Loadit 或 自定义; Xtickformat = 'C'
格式化轴
'(A1)'; 隐藏坐标轴
强制数值 -> NaN.
极坐标: / polar, 转换: = CV-coord (From, Polar = Polar-coord, / to, RECT).

硬拷贝: Set-plot, 'PS' 或 'win' -> Device, / done-file
IDL 编程: 批处理, 宏程序, 过程, 函数, 子程序 / return 语句通顺.
P173 / P170 (function)
无通用 ~ 前序变量 ~ 后序变量 -> 看 Cd, ~ (P291)
P294 / 默认值 / 关键字 (同图)

判断: 奇数, 非 0 的浮点型, 整 & 实型; 非 0 的 Float, Double & Complex
冲电化奇数; Bool 逻辑起名为 True.
Goto, xx: ... | on_IOError, x (输入 / 右有错误时的跳转) | On-Error (运行错误的跳转)
catch, var (相当于 err 出错 -> 0) @ 例名

上注: .reset-session (P10)
charsize, charthick, xthick, ythick, thick.
X/Y style; 共用轴.
箭头, arrows.
自定义颜色: tulet
各图同轴 / 150

PS 格式: 创建时的标准调用.
CIntf 逻辑 & 生成.
!title, v margin 对页数: 倒回去.
换行: !C