
Subject: Re: RFC 1: Common functions for beginners
Posted by [condor](#) on Mon, 19 May 2003 23:35:49 GMT
[View Forum Message](#) <> [Reply to Message](#)

Ed Wright <ed.wright@jpl.nasa.gov> wrote in message
news:<BAE813AB.B700%ed.wright@jpl.nasa.gov>...

> To: IDL guri
> From: Ed Wright
>
> I have a request for comments (just like the IETF).
>
> I learned the basics of IDL use and programming over a long time interval
> while writing the magnum opus of dlms. Rather than have new users repeat my
> learning process, I'd like your suggestions for the twenty or so most common
> IDL functions a new user should understand.
>
> I mean most basic-most used. Open file, read numbers, plot numbers, write
> text, close file.
>
> As always,
> Ed Wright

I think such alist would depend on the background of the user. Someone mentioned that IDL counts from zero, which is important if you have a fortran background but natural to someone who comes from C. (and could go either way with someone who doesn't know any programming at all). The 'format' for formatted output is trivial to the fortran guy, but will stump the C-programmer. etc.

I wonder why nobody has mentioned indgen/findgen yet.

I also don't think anybody has mentioned plotting into a PS device.

Someone needs to spell out that there's two "help-functions", one is called "?" and the other is called "help". Something like "help,IP,/struct" can tell you where to start looking for something or what to look for in the "?".

If people have any programming background at all, I'd hand them "begin...end" as the basic grouping tool and mention that "end" can often optionally be replaced with a more descriptive "endfor" or "endif" and such and that this should be done. (I spent a year or two blindly using 'if ... begin ... endif else begin ... endelse' without knowing why IDL syntax for 'else' was so screwy, until someone told me to think of the 'endif' and 'endelse' as 'end' and suddenly it's all fairly trivial).

Integers are 16 bit! "Loop limit expression too large". 'Nuff said.

Many people mentioned "where" and it should include ",/count" so that "if count gt 0 then ..."

The most common things in PLOT: psym=, thick=,
[xy]range=,[xy]style,color=
[xy]title= vs. title= vs. subtitle=

```
a = indgen(20) & a[0:10] = (b = (a[5:15] = indgen(11)*3)) & print,a  
; etc
```

When to ignore error messages (like Floating errors and such)

$a = b + c/d$; if b,c and d are arrays of different dimensions.

Some word of warning about 'histogram'

I disagree that people need to know "device,decomposed=0" as this already presumes a certain color model. Instead I'd introduce to people the different color models and then direct them to something that works for their machine (Some students are being placed in front of a 5-year old Sparc box with 256 colors, some others have 24 (or even 32-bit displays...) PLUS Some method to get a few clear, well-known colors (red, green, blue, yellow) into their plot on screen and in PS

Just what comes to my mind...
