Subject: Easy question?
Posted by colinr on Wed, 14 Jul 1999 07:00:00 GMT
View Forum Message <> Reply to Message

If I have an i*j*k array and vector of length k and I want to
put the vector in every column of the array how do I do it?

More specifically, I have a 3-d array and wish to compute the average
value of the array over two of the dimensions and store the result in
a 3-d array of the same size as the original.

--
Colin Rosenthal
Astrophysics Institute
University of Oslo

Subject: Re: easy question
Posted by Haje Korth on Thu, 06 Oct 2005 18:21:08 GMT
View Forum Message <> Reply to Message

use "case 1of"

<ChiChiRuiz@gmail.com> wrote in message
news:1128621045.617508.124150@g47g2000cwa.googlegroups.com.. .
> I'm pretty new to IDL. I've learned that the if-else statement goes
> like this:
> if (expression) then begin
> ...
> endif else begin
> ...
> endelse
>
> I was wondering what if IDL supports the following (as in Matlab):
> if (expression1) then ...
> else if (expression2) then...
> else if (expression3) then...
> ...
>
> endif
>
> What's an alternative approach for it? Or I'll need to write a nested
> if-else statement?
> thanks for your help.
>

Subject: Re: easy question
Posted by Michael Wallace on Thu, 06 Oct 2005 18:27:36 GMT
View Forum Message <> Reply to Message

You can use a switch or case statement in this case. You can do
something like the following

```
case 1 of
  (x eq 1): print, 'Test 1'
  (x eq 2): print, 'Test 2'
  (x eq 3): print, 'Test 3'
  (x eq 4): print, 'Test 4'
endcase
```

I'm looking for the first expression that evaluates to true (1). Each
of the expressions are evaluated in turn and the statement associated
with the first expression that evaluates to true is run. If this same
logic were written in a language such as C, it would look like this:

```
if (x == 1) printf("Test 1\n")
else if (x == 2) printf("Test 2\n")
else if (x == 3) printf("Test 3\n")
else if (x == 4) printf("Test 4\n");
```

The case construct I used at the top doesn't exist in C or other
languages because there's usually a requirement that the values in the
case or switch must be constants rather than expressions.

Hope that helps,
-Mike

Subject: Re: easy question
Posted by ChiChiRuiz@gmail.com on Thu, 06 Oct 2005 21:55:29 GMT
View Forum Message <> Reply to Message

thanks a bunch!

Subject: Re: easy question
Posted by mperrin+news on Fri, 07 Oct 2005 02:48:24 GMT
View Forum Message <> Reply to Message

Both Michael and Haje are correct that one can use a case statement
for this sort of thing. But IDL handles a series of if statements just
fine, too:

```
 pro demo, arg
```

```
if arg eq 1 then begin
   print,"one"
endif else if arg eq 2 then begin
   print,"two"
endif else begin
   print, "many"
endelse

end
```

This requires typing a lot more characters than the case version,
but I think it can sometimes make the program logic a bit clearer.
Either way works, so feel free to use whichever idiom you feel more
comfortable with.

- Marshall

Subject: Re: easy question
Posted by Richard French on Fri, 07 Oct 2005 19:58:18 GMT
View Forum Message <> Reply to Message

You can also dispense with the begin and endif/endelse entries as follows:

```
pro mydemo,arg

if arg eq 1 then print,"one" $
      else if arg eq 2 then print,"two" $
      else print, "many
end

mydemo,1
mydemo,2
mydemo,3
End
```

On 10/6/05 10:48 PM, in article di4nlo$2550$1@agate.berkeley.edu, "Marshall
Perrin" <mperrin+news@cymric.berkeley.edu> wrote:

>
> Both Michael and Haje are correct that one can use a case statement
> for this sort of thing. But IDL handles a series of if statements just
> fine, too:
>

```
>  pro demo, arg
>
>  if arg eq 1 then begin
>     print,"one"
>  endif else if arg eq 2 then begin
>     print,"two"
>  endif else begin
>     print, "many"
>  endelse
>
>  end
>
>  This requires typing a lot more characters than the case version,
>  but I think it can sometimes make the program logic a bit clearer.
>  Either way works, so feel free to use whichever idiom you feel more
>  comfortable with.
>
>   - Marshall
>
>
```
