Subject: Re: Do you find this weird too?
Posted by Allan Whiteford on Mon, 19 Nov 2007 11:01:00 GMT
View Forum Message <> Reply to Message

Lasse Clausen wrote:
> On 19 Nov, 10:36, Allan Whiteford
> <allan.rem...@phys.remove.strath.ac.remove.uk> wrote:
>
>> Lasse,
>>
>> From "Using arrays as subscripts":
>>
>> "Clipping
>> If an element of the subscript array is less than or equal to zero, the
>> first element of the subscripted array is selected. If an element of the
>> subscript array is greater than or equal to the last subscript in the
>> subscripted array, the last element is selected. "
>>
>> this is what it's supposed to do although I've never found any kind of
>> use for it.
>>
>> You can say scary things like:
>>
>> a=findgen(50)
>> a[[100]]=7   ; not the same as a[100]=7
>> print,a[49]
>>
>> for me, it throws away all the nice error checking you thought you had
>> for arrays overrunning.
>>
>> Thanks,
>>
>> Allan
>>
>> Lasse Clausen wrote:
>>
>>> Hi there,
>>
>>> hope this gets through to you guys, now with all the spam and whatnot.
>>> Anywho, maybe this has come up before, here is some code
>>
>>> aa = randomu(12L, 200, 100)
>>> ff = findgen(100)
>>> maxs = max(aa, maxind, dimension=2)
>>> print, maxind[0:10]
>>> help, ff[maxind[0:10]]
>>> plot, ff[maxind[0:10]]
>>

>>> Now I find this weird, because maxind is an array of longs clearly
>>> bigger than the size of ff but IDL does not complain and plots
>>> something (btw not what I want but this is solved with array_indices).
>>> Is this something to do with the fact that maxind is a 1D
>>> representation of 2D array indices? I hope it is because otherwise why
>>> does IDL not fall over complaining that maxind[0], which on my machine
>>> is 19200, is bigger than the size of ff.
>>
>>> Mhmm
>>
>>> print, ff[maxind[0]]
>>
>>> falls over
>>
>>> print, ff[maxind[0:1]]
>>
>>> doesn't.
>>
>>> btw print, !version
>>> { x86 linux unix linux 6.2 Jun 20 2005   32   64}
>>
>>> Cheers
>>> Lasse
>
>
> What the...?! Well, I guess somebody thought this was a good idea.
> Makes no sense to me since you could achieve the same behaviour using
> the < or > operator. Subscripting an array with an index smaller than
> zero or bigger than the size of the array is just wrong and should
> produce an error.
>
> But thanks, I will keep this useful fact in mind.
>
> Cheers
> Lasse

Lasse,

I agree completely.

It's really scary: now you need to go back and check all the code you thought was correct because IDL throws errors on bad subscript values so no need to check for errors there... or so you thought!

I guess it's hard to change now since legacy code would break (although, I rather suspect any code that breaks would be because it's always been wrong rather than because it's relying on this behaviour).

A compile option to turn this behaviour off might be a good compromise.

Can anyone post an example of code where this feature is useful?

Thanks,

Allan