Subject: Re: Access array elements with String
Posted by Bob[3] on Mon, 14 Jul 2008 19:23:40 GMT
View Forum Message <> Reply to Message

On Jul 14, 1:06 pm, humanumbre...@gmail.com wrote:
> On Jul 14, 12:59 pm, humanumbre...@gmail.com wrote:
>
>
>
>
>
>> On Jul 14, 12:30 pm, Bob Crawford <Snowma...@gmail.com> wrote:
>
>>> On Jul 14, 11:49 am, humanumbre...@gmail.com wrote:
>
>>>> On Jul 14, 11:41 am, Bob Crawford <Snowma...@gmail.com> wrote:
>
>>>> > On Jul 14, 11:16 am, humanumbre...@gmail.com wrote:
>
>>>> > > Hello all,
>
>>>> > > Another issue - perhaps one of you has encountered this before. It's
>>>> > > sort of a neat problem. I'm attempting to build array subscripts on
>>>> > > the fly based on user input. IE the number of static/variable elements
>>>> > > is changing, which allows the user to pick different axes to plot.
>>>> > > Nevermind all that.
>
>>>> > > Anyway, let's say a user wants a particular axis to be variable. In
>>>> > > this case, the dataset array where I'm attempting to pull values from
>>>> > > would contain a *, to get all these elements. Unfortunately, I do not
>>>> > > know in advance which dimension of the array I will be using, so I am
>>>> > > attempting to build the subscript based on a string.
>
>>>> > > This was my original thought:
>>>> > > a = dindgen(5,5,5)
>>>> > > b = ['3','3','3']
>>>> > > print, a[b]
>>>> > > but this just returns a[3], a[3], a[3]
>
>>>> > > So, I figured I'd do it this way:
>>>> > > c = '3'
>>>> > > print, a[c,c,c] -- This works!
>
>>>> > > Now for the gold,
>>>> > > d = '*'
>>>> > > print, a[c,c,d] -- error - can't convert string-> long
>>>> > > so I get an idea-- maybe I'll just use the ascii value for the
>>>> > > asterisk.

>>>> > > d = String(42b)
>>>> > > print, a[c,d,d] -- error - can't convert string-> long
>
>>>> > > Any thoughts ?
>>>> > > Thanks in advance
>>>> > > --Justin
>
>>>> > Why try to force the '*' - might not SIZE be more useful?
>>>> > e.g.
>>>> > s=SIZE(a)
>>>> > print, a[c,c,s[3]] ; for a[c,c,d]
>>>> > print, a[c,s[2],s[3]]; for a[c,d,d]
>
>>>> Hey Bob,
>
>>>> Thanks for the post!
>>>> I think I may need to elaborate a bit more --
>>>> I need the entire row of the multi-dimensional array.
>>>> So, for example, let's say I have an array that is 30 x 20 x 50
>>>> I will need *,0,0 to plot the first 30 values
>>>> but I could just as easily need 0,*,0 or 0,0,* Depending on user
>>>> input, so I can't anticipate that in advance.
>
>>>> Cheers,
>>>> --Justin- Hide quoted text -
>
>>>> - Show quoted text -
>
>>> Oops.
>>> I posted too soon (thank you for the clarification Justin - that is
>>> what I was trying to do)
>>> Here is what I should have posted:
>
>>> print, a[c,c,0:(s[3]-1)] ; for a[c,c,d]
>>> print, a[c,0:(s[2]-1),0:(s[3]-1)]; for a[c,d,d]
>
>>> Isn't '*' just short form notation for 0:(s[n]-1), anyway?
>
>> Hey Bob,
>
>> Yes, I think '*' is short for 0:(s[n]-1) but I read somewhere that you
>> shouldn't use the range because of performance issues...
>> Can anyone shed light on that issue?
>
> IE:
> From the "Help" pages on "Arrays"
> "Processing subscript ranges is inefficient. When possible, use an
> array or scalar subscript instead of specifying a subscript range

> where the beginning and ending subscripts are separated by the colon
> character."- Hide quoted text -
>
> - Show quoted text -

It appears from my reading of David's page that quoting the subscripts
by way of using a colon is not any different (memory wise) than using
a '*'.

Given the passage you've quoted above doesn't state that using a '*'
is any more efficient than ':' subsetting, but perhaps doing (as
suggested):

```
b=INDGEN(s(3))
print, a[c,c,b]
```

might be more efficient - dunno, haven't tested.