Subject: Loop through arrays with different names
Posted by hazel on Thu, 02 Aug 2007 00:04:40 GMT
View Forum Message <> Reply to Message

Hello,

I have a really simple problem that always happens when I starting trying to use strings..

I want to loop through a series of arrays, each with a slightly different name (det1arr(n,3), det2arr(n,3), det3arr(n,3)..etc..) and subject each to the same function:

a quick outline is:

for j=1,8 do begin

for i=0, n-1 do begin

vx=det1arr(i, 0) ; where the 1 is effectively the value of 'j'
vy=det1arr(i, 1)
vz=det1arr(i, 2)

result(i,j)=some_function(vx, vy, vz)

endfor

So can I do something like: vx=det+j+arr(i,0)? (Only that clearly won't work)? There must be a way? Otherwise I can reload all my different arrays into one array I realise and just loop through that.

I'm sure it's something painfully simple and for that I apologise, but it is the end of the day (for me anyway),

Thanks,

Hazel

Subject: Re: Loop through arrays with different names
Posted by MarioIncandenza on Thu, 02 Aug 2007 02:09:36 GMT
View Forum Message <> Reply to Message

On Aug 1, 5:04 pm, hazel <hazel.ken...@gmail.com> wrote:
> So can I do something like: vx=det+j+arr(i,0)? (Only that clearly
> Otherwise I can reload all my
> different arrays into one array I realise and just loop through that.

What you want to do, if it's at all practical, is put everything into
a single 2D array and access it that way. If that's not practical, you
can do "kludge a" or "kludge b":

```
;kludge a
IDL> detarrsp=ptrarr(8)
IDL> detarrsp[0]=ptr_new(det1arr)
IDL> detarrsp[1]=ptr_new(det2arr)
; etc.
IDL> for j=0,7 do for i=0,n-1 do
result[i,j]=some_function((*detarrsp[j])[i,0:2])


; kludge b
IDL> for j=1,8 do for i=0,n-1 do success_or_failure= $
    execute(string(i,j,j,i,format= $
    '("result[",i,",",j,"]=some_function(det",i1.1,"arr[",i, ",
0:2])")'))
```

"Kludge B" is only for the most desperate of straits, and if it is
possible to put the data into a 2D array, you might as well avoid
"kludge a" as well.

Good luck,

Edward H.

Subject: Re: Loop through arrays with different names
Posted by Foldy Lajos on Thu, 02 Aug 2007 07:03:12 GMT
View Forum Message <> Reply to Message

On Thu, 2 Aug 2007, hazel wrote:

> So can I do something like: vx=det+j+arr(i,0)? (Only that clearly
> won't work)? There must be a way? Otherwise I can reload all my
> different arrays into one array I realise and just loop through that.

```
vx=(scope_varfetch('det'+strtrim(string(j), 1)+'arr'))[i,0]
```

regards,
lajos

Subject: Re: Loop through arrays with different names
Posted by mattf on Thu, 02 Aug 2007 12:57:54 GMT

On Aug 1, 8:04 pm, hazel <hazel.ken...@gmail.com> wrote:
> Hello,
>
> I have a really simple problem that always happens when I starting
> trying to use strings..
>
> I want to loop through a series of arrays, each with a slightly
> different name (det1arr(n,3), det2arr(n,3), det3arr(n,3)..etc..) and
> subject each to the same function:
>
> a quick outline is:
>
> for j=1,8 do begin
>
> for i=0, n-1 do begin
>
> vx=det1arr(i, 0) ; where the 1 is effectively the value of 'j'
> vy=det1arr(i, 1)
> vz=det1arr(i, 2)
>
> result(i,j)=some_function(vx, vy, vz)
>
> endfor
>
> So can I do something like: vx=det+j+arr(i,0)? (Only that clearly
> won't work)? There must be a way? Otherwise I can reload all my
> different arrays into one array I realise and just loop through that.
>
> I'm sure it's something painfully simple and for that I apologise, but
> it is the end of the day (for me anyway),
>
> Thanks,
>
> Hazel

A somewhat more flexible approach is to collect the similar variables
into a struct:

det_struct = {d1:det1arr, d2:det2arr, ...}

and then do a for-loop over the struct's fields

```
n_fields = n_tags(det_struct)
for ix_struct = 0, n_fields-1 do begin
  current_field = det_struct.(ix)
  for ix_var = 0, n do begin
     -- do whatever here --
```

endfor
endfor

Subject: Re: Loop through arrays with different names
Posted by mattf on Thu, 02 Aug 2007 13:00:27 GMT
View Forum Message <> Reply to Message

On Aug 2, 8:57 am, mattf <mfe...@earthlink.net> wrote:
> On Aug 1, 8:04 pm, hazel <hazel.ken...@gmail.com> wrote:
>
>
>
>> Hello,
>
>> I have a really simple problem that always happens when I starting
>> trying to use strings..
>
>> I want to loop through a series of arrays, each with a slightly
>> different name (det1arr(n,3), det2arr(n,3), det3arr(n,3)..etc..) and
>> subject each to the same function:
>
>> a quick outline is:
>
>> for j=1,8 do begin
>
>> for i=0, n-1 do begin
>
>> vx=det1arr(i, 0) ; where the 1 is effectively the value of 'j'
>> vy=det1arr(i, 1)
>> vz=det1arr(i, 2)
>
>> result(i,j)=some_function(vx, vy, vz)
>
>> endfor
>
>> So can I do something like: vx=det+j+arr(i,0)? (Only that clearly
>> won't work)? There must be a way? Otherwise I can reload all my
>> different arrays into one array I realise and just loop through that.
>
>> I'm sure it's something painfully simple and for that I apologise, but
>> it is the end of the day (for me anyway),
>
>> Thanks,
>
>> Hazel
>
> A somewhat more flexible approach is to collect the similar variables

> into a struct:
>
> det_struct = {d1:det1arr, d2:det2arr, ...}
>
> and then do a for-loop over the struct's fields
>
> n_fields = n_tags(det_struct)
> for ix_struct = 0, n_fields-1 do begin
> current_field = det_struct.(ix)
> for ix_var = 0, n do begin
> -- do whatever here --
> endfor
> endfor

Needless to say, I got the critical line wrong-- it should be

current_field = det_struct.(ix_struct)

Subject: Re: Loop through arrays with different names
Posted by greg.addr on Thu, 02 Aug 2007 13:27:55 GMT
View Forum Message <> Reply to Message

On Aug 2, 2:04 am, hazel <hazel.ken...@gmail.com> wrote:
> Hello,
>
> I have a really simple problem that always happens when I starting
> trying to use strings..
>
> I want to loop through a series of arrays, each with a slightly
> different name (det1arr(n,3), det2arr(n,3), det3arr(n,3)..etc..) and
> subject each to the same function:
>
> a quick outline is:
>
> for j=1,8 do begin
>
> for i=0, n-1 do begin
>
> vx=det1arr(i, 0) ; where the 1 is effectively the value of 'j'
> vy=det1arr(i, 1)
> vz=det1arr(i, 2)
>
> result(i,j)=some_function(vx, vy, vz)
>
> endfor
>
> So can I do something like: vx=det+j+arr(i,0)? (Only that clearly

> won't work)? There must be a way? Otherwise I can reload all my
> different arrays into one array I realise and just loop through that.
>
> I'm sure it's something painfully simple and for that I apologise, but
> it is the end of the day (for me anyway),
>
> Thanks,
>
> Hazel

I'd vote for putting them in a single array. I'm pretty sure this will
turn out the cleanest option. You can put them together like this:

detarr=[[[det1arr]],[[det2arr]],...]

although it would likely be better to create them in a detarr(n,3,8)
in the first place. You may find that you can carry out your entire
calculation as an operation on the array, rather than chugging through
all the elements.

regards,
greg

Subject: Re: Loop through arrays with different names
Posted by hazel on Thu, 02 Aug 2007 15:55:58 GMT
View Forum Message <> Reply to Message

On Aug 2, 7:27 am, greg.a...@googlemail.com wrote:
> On Aug 2, 2:04 am, hazel <hazel.ken...@gmail.com> wrote:
>
>
>
>> Hello,
>
>> I have a really simple problem that always happens when I starting
>> trying to use strings..
>
>> I want to loop through a series of arrays, each with a slightly
>> different name (det1arr(n,3), det2arr(n,3), det3arr(n,3)..etc..) and
>> subject each to the same function:
>
>> a quick outline is:
>
>> for j=1,8 do begin
>
>> for i=0, n-1 do begin
>

>> vx=det1arr(i, 0) ; where the 1 is effectively the value of 'j'
>> vy=det1arr(i, 1)
>> vz=det1arr(i, 2)
>
>> result(i,j)=some_function(vx, vy, vz)
>
>> endfor
>
>> So can I do something like: vx=det+j+arr(i,0)? (Only that clearly
>> won't work)? There must be a way? Otherwise I can reload all my
>> different arrays into one array I realise and just loop through that.
>
>> I'm sure it's something painfully simple and for that I apologise, but
>> it is the end of the day (for me anyway),
>
>> Thanks,
>
>> Hazel
>
> I'd vote for putting them in a single array. I'm pretty sure this will
> turn out the cleanest option. You can put them together like this:
>
> detarr=[[[det1arr]],[[det2arr]],...]
>
> although it would likely be better to create them in a detarr(n,3,8)
> in the first place. You may find that you can carry out your entire
> calculation as an operation on the array, rather than chugging through
> all the elements.
>
> regards,
> greg

Thank-you all for your useful suggestions, problem solved!

Hazel