
Subject: turn a string value into variable name

Posted by [Nick\[2\]](#) on Thu, 23 Oct 2008 22:30:54 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hello,

I am restoring over 100 .sav files and creating structures for analysis. Each .sav represents a different scenario and ideally I would like to name the created structure in sequential format (case1, case2...casen).

To pull the data in I am using a for loop (all .sav have identical array names) and I'd like to use the counter to name the variable. Is it possible to turn a string into a variable name?

Thanks,

Nick

Subject: Re: turn a string value into variable name

Posted by [David Fanning](#) on Wed, 29 Oct 2008 13:34:29 GMT

[View Forum Message](#) <> [Reply to Message](#)

loebasboy writes:

> On 23 okt, 23:56, David Fanning <n...@dfanning.com> wrote:

>> Nick writes:

>>> I am restoring over 100 .sav files and creating structures for
>>> analysis. Each .sav represents a different scenario and ideally I
>>> would like to name the created structure in sequential format (case1,
>>> case2...casen).

>>

>>> To pull the data in I am using a for loop (all .sav have identical
>>> array names) and I'd like to use the counter to name the variable. Is
> s

>>> it possible to turn a string into a variable name?

>>

>> The EXECUTE command can do this:

>

> Thank you David, I had the same question !

I'm glad the answer is helpful, but I would say in general that 9 people out of 10 who are writing code like this are *probably* doing the wrong thing. That is to say, the good reasons for creating variables on the fly, inside a procedure, are few and far between. I'd look pretty hard for alternatives before I ever coded something

like this up.

Cheers,

David

--

David Fanning, Ph.D.

Fanning Software Consulting, Inc.

Coyote's Guide to IDL Programming: <http://www.dfanning.com/>

Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: turn a string value into variable name

Posted by [loebasboy](#) on Wed, 29 Oct 2008 14:55:16 GMT

[View Forum Message](#) <> [Reply to Message](#)

On 29 okt, 14:34, David Fanning <n...@dfanning.com> wrote:

> loebasboy writes:

>> On 23 okt, 23:56, David Fanning <n...@dfanning.com> wrote:

>>> Nick writes:

>>>> I am restoring over 100 .sav files and creating structures for

>>>> analysis. =A0Each .sav represents a differencet scenario and ideally I

>>>> would like to name the created structure in sequential format (case1,

>>>> case2...casen).

>

>>>> To pull the data in I am using a for loop (all .sav have identical

>>>> array names) and I'd like to use the counter to name the variable. =A0I=

>> s

>>>> it possible to turn a string into a variable name?

>

>>> The EXECUTE command can do this:

>

>> Thank you David, I had the same question !

>

> I'm glad the answer is helpful, but I would say in general

> that 9 people out of 10 who are writing code like this

> are *probably* doing the wrong thing. That is to say,

> the good reasons for creating variables on the fly,

> inside a procedure, are few and far between. I'd look

> pretty hard for alternatives before I ever coded something

> like this up.

>

> Cheers,

>

> David

>

> --

- > David Fanning, Ph.D.
- > Fanning Software Consulting, Inc.
- > Coyote's Guide to IDL Programming:<http://www.dfanning.com/>
- > Sepore ma de ni thui. ("Perhaps thou speakest truth.")- Tekst uit oorspronkelijk bericht niet weergegeven -
- >
- > - Tekst uit oorspronkelijk bericht weergegeven -

To know if my code is good coding I'll explain myself why I want to use this type of coding.

I want to write function to read data out of a HDF5 file with a certain structure. The HDF5 file consists of a digital aerial image with callibration information per band and such. What I wanted to do is that I want to read in the central wavelength of a band, analyse it and define the name of that band so that I can write the data of the band in a variable with the right name of that band. Lets say that the central wavelength of this band is 752 nm, I would like to write the information of this band in the variable "NIR". So I first read in the central wavelength of a certain band, compare it with a reference value (depends on the sensor), and then write the information of the band in the right named variable. In this way I could do analyses with logical named variables...for instance, the calculation of the NDVI simply becomes (NIR-R)/(NIR+R) in the code.

The code:

```

FOR m = 0, nr_bands[0]-1 DO BEGIN
    ctrlwav_str = string("/SensorData/Band",string(m+1,
format='(I03)'),"/SpectralResponse/CentralWavelength")
    dataset_id = H5D_OPEN(file_id, ctrlwav_str)
    ctrlwav[m] = H5D_READ(dataset_id)
    H5D_CLOSE, dataset_id
    IF (abs(ctrlwav[m] - color_id[0]) LT 0.015) THEN
col_band[m] = "PAN"
    IF (abs(ctrlwav[m] - color_id[1]) LT 0.015) THEN
col_band[m] = "B"
    IF (abs(ctrlwav[m] - color_id[2]) LT 0.015) THEN
col_band[m] = "G"
    IF (abs(ctrlwav[m] - color_id[3]) LT 0.015) THEN
col_band[m] = "R"
    IF (abs(ctrlwav[m] - color_id[4]) LT 0.015) THEN
col_band[m] = "NIR"
    data_str = string("/SensorData/Band",string(m+1,
format='(I03)'),"/SensorData")
    dataset_id = H5D_OPEN(file_id, data_str)
    tmp = Execute(col_band[m] + ' = H5D_READ(dataset_id)')
    H5D_CLOSE, dataset_id

```

ENDFOR

Subject: Re: turn a string value into variable name

Posted by [David Fanning](#) on Wed, 29 Oct 2008 15:09:33 GMT

[View Forum Message](#) <> [Reply to Message](#)

loebasboy writes:

> To know if my code is good coding I'll explain myself why I want to
> use this type of coding.
>
> I want to write function to read data out of a HDF5 file with a
> certain structure. The HDF5 file consists of a digital aerial image
> with calibration information per band and such. What I wanted to do
> is that I want to read in the central wavelength of a band, analyse it
> and define the name of that band so that I can write the data of the
> band in a variable with the right name of that band. Lets say that the
> central wavelength of this band is 752 nm, I would like to write the
> information of this band in the variable "NIR". So I first read in the
> central wavelength of a certain band, compare it with a reference
> value (depends on the sensor), and then write the information of the
> band in the right named variable. In this way I could do analyses with
> logical named variables...for instance, the calculation of the NDVI
> simply becomes (NIR-R)/(NIR+R) in the code.

Well, I think I'll just stand behind my statement. :-)

Cheers,

David

P.S. Wouldn't just using simple variables make your program easier
to understand!? That's sort of the purpose of variables, to
stand in for the real thing.

--

David Fanning, Ph.D.

Fanning Software Consulting, Inc.

Coyote's Guide to IDL Programming: <http://www.dfanning.com/>

Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: turn a string value into variable name

Posted by [loebasboy](#) on Wed, 29 Oct 2008 15:29:17 GMT

[View Forum Message](#) <> [Reply to Message](#)

On 29 okt, 16:09, David Fanning <n...@dfanning.com> wrote:

> loebasboy writes:

>> To know if my code is good coding I'll explain myself why I want to

>> use this type of coding.

>

>> I want to write function to read data out of a HDF5 file with a

>> certain structure. The HDF5 file consists of a digital aerial image

>> with callibration information per band and such. What I wanted to do

>> is that I want to read in the central wavelength of a band, analyse it

>> and define the name of that band so that I can write the data of the

>> band in a variable with the right name of that band. Lets say that the

>> central wavelength of this band is 752 nm, I would like to write the

>> information of this band in the variable "NIR". So I first read in the

>> central wavelength of a certain band, compare it with a reference

>> value (depends on the sensor), and then write the information of the

>> band in the right named variable. In this way I could do analyses with

>> logical named variables...for instance, the calculation of the NDVI

>> simply becomes (NIR-R)/(NIR+R) in the code.

>

> Well, I think I'll just stand behind my statement. :-)

>

> Cheers,

>

> David

>

> P.S. Wouldn't just using simple variables make your program easier

> to understand!? That's sort of the purpose of variables, to

> stand in for the real thing.

>

> --

> David Fanning, Ph.D.

> Fanning Software Consulting, Inc.

> Coyote's Guide to IDL Programming:<http://www.dfanning.com/>

> Sepore ma de ni thui. ("Perhaps thou speakest truth.")- Tekst uit oorspronkelijk bericht niet weergegeven -

>

> - Tekst uit oorspronkelijk bericht weergegeven -

simple AND logical named variables? ;)

Subject: Re: turn a string value into variable name

Posted by [Vince Hradil](#) on Wed, 29 Oct 2008 15:30:30 GMT

[View Forum Message](#) <> [Reply to Message](#)

On Oct 29, 10:29 am, loebasboy <stijn....@gmail.com> wrote:

> On 29 okt, 16:09, David Fanning <n...@dfanning.com> wrote:

>

```

>
>
>> loebasboy writes:
>>> To know if my code is good coding I'll explain myself why I want to
>>> use this type of coding.
>
>>> I want to write function to read data out of a HDF5 file with a
>>> certain structure. The HDF5 file consists of a digital aerial image
>>> with calibration information per band and such. What I wanted to do
>>> is that I want to read in the central wavelength of a band, analyse it
>>> and define the name of that band so that I can write the data of the
>>> band in a variable with the right name of that band. Lets say that the
>>> central wavelength of this band is 752 nm, I would like to write the
>>> information of this band in the variable "NIR". So I first read in the
>>> central wavelength of a certain band, compare it with a reference
>>> value (depends on the sensor), and then write the information of the
>>> band in the right named variable. In this way I could do analyses with
>>> logical named variables...for instance, the calculation of the NDVI
>>> simply becomes (NIR-R)/(NIR+R) in the code.
>
>> Well, I think I'll just stand behind my statement. :-)
>
>> Cheers,
>
>> David
>
>> P.S. Wouldn't just using simple variables make your program easier
>> to understand!? That's sort of the purpose of variables, to
>> stand in for the real thing.
>
>> --
>> David Fanning, Ph.D.
>> Fanning Software Consulting, Inc.
>> Coyote's Guide to IDL Programming: http://www.dfanning.com/
>> Sepore ma de ni thui. ("Perhaps thou speakest truth.")- Tekst uit oorspronkelijk bericht niet
weergegeven -
>
>> - Tekst uit oorspronkelijk bericht weergegeven -
>
> simple AND logical named variables? ;)

```

<http://en.wikipedia.org/wiki/Obfuscated>

Subject: Re: turn a string value into variable name
 Posted by [laura.hike](#) on Thu, 01 Jun 2017 22:48:20 GMT
[View Forum Message](#) <> [Reply to Message](#)

So now I want to do this in order to use a loop to add lines to a plot and then make a legend. I need the plot names to all be different so I can use them in the "target" argument, but I can otherwise make the plot commands identical, e.g.,

```
p1 = plot(xbin, pdf, color = linecolors[nsea], thick = 2, name = seanames[nsea], /overplot)
```

Based on David's comment, I tried

```
execute('p5 = plot(xbin, pdf, color = linecolors[nsea-1], thick = 2, name = seanames[nsea-1], /overplot)')
```

but this didn't do anything except return a "1". I'm really not excited about putting in a plot command (with an 'if' statement' for each of the lines I want to overplot.

Subject: Re: turn a string value into variable name

Posted by [Markus Schmassmann](#) on Fri, 02 Jun 2017 09:59:53 GMT

[View Forum Message](#) <> [Reply to Message](#)

On 06/02/2017 12:48 AM, laura.hike@gmail.com wrote:

> So now I want to do this in order to use a loop to add lines to a plot and then make a legend. I need the plot names to all be different so I can use them in the "target" argument, but I can otherwise make the plot commands identical, e.g.,

>

> p1 = plot(xbin, pdf, color = linecolors[nsea], thick = 2, name = seanames[nsea], /overplot)

>

> Based on David's comment, I tried

>

> execute('p5 = plot(xbin, pdf, color = linecolors[nsea-1], thick = 2, name = seanames[nsea-1], /overplot)')

>

> but this didn't do anything except return a "1". I'm really not excited about putting in a plot command (with an 'if' statement' for each of the lines I want to overplot.

I don't see why you'd use an EXECUTE here, try the code below

```
pp=objarr(n_lines)
for nsea=0,n_lines-1 do if <condition> then pp[nsea]=plot( xbin, pdf, $
    color = linecolors[nsea-1], thick = 2, name = seanames[nsea-1], $
    /overplot )
l=legend(target=pp)
```

If this doesn't help, please explain in more detail what you want to do, it's now fully clear from your post.

Good Luck, Markus

PS: The main topic of this thread is mostly obsolete as of IDL 8.0 with

the introduction of HASH
