
Subject: Re: code snippet

Posted by [Dick Jackson](#) on Tue, 29 May 2001 21:21:50 GMT

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

Hi,

Ken Mankoff wrote...

```
> Hi everyone,
>
> i just ran across this in an IDL (its inside 'map_continents.pro')
> procedure, and thought its behaviour was pretty neat. [...]
>
> IDL> key = 2
> IDL> ;; set 'key' to 0 or 1 or 2 or ...
> IDL> possibilities = ['first','second','third','fourth']
> IDL> choice = (possibilities)[key]
> IDL> help, choice
> CHOICE      STRING    = 'third'
```

Perhaps less neat is the (equivalent) simple array access:

```
choice = (possibilities)[key]
```

But neater still is subscripting an array built on-the-spot
(again equivalent to the original):

```
choice = ([ 'first','second','third','fourth' ])[key]
```

Note that all parentheses and brackets are required here.

Cheers,

--

-Dick

Dick Jackson / dick@d-jackson.com
D-Jackson Software Consulting / http://www.d-jackson.com
Calgary, Alberta, Canada / +1-403-242-7398 / Fax: 241-7392

Subject: Re: code snippet

Posted by [Craig Markwardt](#) on Tue, 29 May 2001 22:03:48 GMT

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

"Dick Jackson" <dick@d-jackson.com> writes:

```
> choice = ([ 'first','second','third','fourth' ])[key]
```

Heh heh, and just to be archane, why not try these:

```
choice = ([ 'first','second','third','fourth'])(key)
choice = ([ 'first','second','third','fourth'])([key])
choice = ([ 'first','second','third','fourth'])[(key)]
```

One of these four expressions has a slightly different meaning than the other. As a trivia test to the newer users of IDL on the newsgroup (or the older ones!), can you say which one is different? [And why ?]

Craig

--

Craig B. Markwardt, Ph.D. EMAIL: craigmnet@cow.physics.wisc.edu
Astrophysics, IDL, Finance, Derivatives | Remove "net" for better response

Subject: Re: code snippet
Posted by [m.hadfield](#) on Wed, 30 May 2001 00:38:48 GMT
[View Forum Message](#) <> [Reply to Message](#)

From: "Craig Markwardt" <craigmnet@cow.physics.wisc.edu>
> choice = (['first','second','third','fourth'])[key]
> choice = (['first','second','third','fourth'])(key)
> choice = (['first','second','third','fourth'])([key])
> choice = (['first','second','third','fourth'])[(key)]
> One of these four expressions has a slightly different meaning than
> the other. As a trivia test to the newer users of IDL on the
> newsgroup (or the older ones!), can you say which one is different?

I think I qualify as one of the older ones.

IMHO you could (and should) eliminate some of the confusion by declaring "compile_opt STRICTARR" or "compile_opt IDL2" in your startup file so that numbers 2 and 3 are illegal.

Now, this

```
choice = ([ 'first','second','third','fourth'])[key]
```

is the same as this

```
choice = ([ 'first','second','third','fourth'])[(key)]
```

and indeed they are both the same as this

```
choice = (['first','second','third','fourth'])((((((key))))))
```

because all those parentheses around "key" have no effect.

But this

```
choice = (['first','second','third','fourth'])[[key]]
```

is different from all the others because the index is now a 1-D array, so the result is also a 1-D array.

But what about these

```
choice = (['first','second','third','fourth'])[[[[[key]]]]]
choice = (['first','second','third','fourth'])[[([([key]))]]]
choice = ([['first','second'], ['third','fourth']])[[key]]
choice = ([['first','second','third'], 'fourth'])[[key]]
```

I'll leave them as an exercise (I think they're all valid).

Mark Hadfield
m.hadfield@niwa.cri.nz <http://katipo.niwa.cri.nz/~hadfield>
National Institute for Water and Atmospheric Research

--

Posted from clam.niwa.cri.nz [202.36.29.1]
via Mailgate.ORG Server - <http://www.Mailgate.ORG>

Subject: Re: code snippet
Posted by [Brian Jackel](#) on Wed, 30 May 2001 16:26:20 GMT
[View Forum Message](#) <> [Reply to Message](#)

But then there's the following fun:

```
IDL> key= 1 & print,['first','second','third','fourth'])[[key]]
second
IDL> key= 3 & print,['first','second','third','fourth'])[[key]]
fourth
IDL> key= 5 & print,['first','second','third','fourth'])[[key]]
fourth
IDL> key= -999 & print,['first','second','third','fourth'])[[key]]
first
```

Which is a "feature" when you expect it, and a bug when you don't :)

Brian
