

---

Subject: Re: Order of argument evaluation (Was: Re: making a checkerboard array?)

Posted by [David Fanning](#) on Wed, 26 Sep 2007 13:17:35 GMT

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

---

Allan Whiteford writes:

```
> For a more reasonable example am I allowed to do:
>
> plot,(x=findgen(11)),x^2
>
> and rely on the first argument of plot being evaluated before the second
> argument (and the side effect of creating 'x') similarly for creating an
> array and editing in place, e.g.:
>
> IDL> ((a=fltarr(3)))[1]=10
> IDL> print,a
>      0.00000    10.0000    0.00000
>
> Does anyone else do this or just me? I tend to only do things like this
> at the command line and don't embed it in programs out of fear that
> it's going to stop working one day. It also arguably makes the code less
> readable but as a pattern it's probably not that bad. I'd like to know
> what others think.
```

Parentheses have the highest order of precedence, so what is inside them always gets done first. Since what is inside them in these cases is a variable creation, I can't see how you could possibly be breaking any rules, and the construction seems legitimate to me.

That said, I am generally in the business of writing understandable code (as opposed to JD, for example, whose business is to write elegant code), and I can just imagine the number of students and clients who will be left scratching their heads over this kind of syntax. Might as well write the darn thing in LISP to start with!

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: Order of argument evaluation (Was: Re: making a checkerboard array?)

Posted by [Allan Whiteford](#) on Wed, 26 Sep 2007 13:37:59 GMT

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

---

David Fanning wrote:

> Allan Whiteford writes:

>

>

>> For a more reasonable example am I allowed to do:

>>

>> plot,(x=findgen(11)),x^2

>>

>> and rely on the first argument of plot being evaluated before the second

>> argument (and the side effect of creating 'x') similarly for creating an

>> array and editing in place, e.g.:

>>

>> IDL> ((a=fltarr(3)))[1]=10

>> IDL> print,a

>> 0.00000 10.0000 0.00000

>>

>> Does anyone else do this or just me? I tend to only do things like this

>> at the command line and don't embed it in programs out of fear that

>> it's going to stop working one day. It also arguably makes the code less

>> readable but as a pattern it's probably not that bad. I'd like to know

>> what others think.

>

>

> Parentheses have the highest order of precedence, so what

> is inside them always gets done first. Since what is inside

> them in these cases is a variable creation, I can't see how

> you could possibly be breaking any rules, and the construction

> seems legitimate to me.

>

How about the order of evaluation of procedure arguments (like the plot example above)? I've never found anything saying it'll evaluate the leftmost argument first and you can rely on its side effects for the arguments on the right. In an extreme example I can put both arguments in parentheses and I still seem to be ok:

```
plot,(x=findgen(11)),(x^2)
```

> That said, I am generally in the business of writing

> understandable code (as opposed to JD, for example, whose

> business is to write elegant code), and I can just imagine

> the number of students and clients who will be left scratching

> their heads over this kind of syntax. Might as well write the

> darn thing is LISP to start with!

>

Well, Eric S Raymond said we should all learn LISP even if we weren't going to use it because it would change our programming styles for the better. I followed his recommendation and it certainly changed my programming style but maybe not for the better!

I'm a bit worried that it's showing so obviously though, you thought of LISP and I had only showed you one line of code.

> Cheers,  
>  
> David

---

---

Subject: Re: Order of argument evaluation (Was: Re: making a checkerboard array?)

Posted by [pgrigis](#) on Wed, 26 Sep 2007 14:18:22 GMT

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

---

Allan Whiteford wrote:

> [...]  
>  
> Anyway, the real point of my post is to ask if I'm allowed to do the  
> above. For a more reasonable example am I allowed to do:  
>  
> plot,(x=findgen(11)),x^2  
>  
> and rely on the first argument of plot being evaluated before the second  
> argument (and the side effect of creating 'x') similarly for creating an  
> array and editing in place, e.g.:  
>  
> IDL> ((a=fltarr(3)))[1]=10  
> IDL> print,a  
>     0.00000     10.0000     0.00000  
>

I guess a danger signal would be appropriate near such statements, after all this can quickly get out of control ;-)

```
IDL> delvar,c
IDL> c[(c=[0,1])]=(c=[1,2,3])
IDL> print,c
   1   1   3
```

Now, go figure this one out...

Ciao,  
Paolo

> Does anyone else do this or just me? I tend to only do things like this  
> at the command line and don't embed it in programs out of fear that  
> it's going to stop working one day. It also arguably makes the code less  
> readable but as a pattern it's probably not that bad. I'd like to know  
> what others think.  
>  
> Thanks,  
>  
> Allan

---

---

Subject: Re: Order of argument evaluation (Was: Re: making a checkerboard array?)

Posted by [Foldy Lajos](#) on Wed, 26 Sep 2007 14:39:05 GMT

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

---

On Wed, 26 Sep 2007, pgrigis@gmail.com wrote:

> Allan Whiteford wrote:  
>> [...]  
>>  
>> IDL> ((a=fltarr(3)))[1]=10  
>> IDL> print,a  
>> 0.00000 10.0000 0.00000  
>>  
>  
> I guess a danger signal would be appropriate  
> near such statements, after all this can quickly  
> get out of control ;-)  
>  
> IDL> delvar,c  
> IDL> c[(c=[0,1])]=(c=[1,2,3])  
> IDL> print,c  
> 1 1 3  
>

In C/C++/Fortran/... there is a rule that a memory cell can be changed at most once in a statement, otherwise the result is undefined (and so different implementations can create different results without violating the standard). I guess a similar rule is in effect in IDL, too.

These examples give syntax error prior IDL 6.0:

```
IDL> print, !version
{ sparc sunos unix 5.3 Nov 11 1999}
IDL> ((a=fltarr(3)))[1]=10 & print, a
% Expression must be named variable in this context: <FLOAT   Array[3]>.
% Execution halted at: $MAIN$
IDL>
```

regards,  
lajos

---

---

Subject: Re: Order of argument evaluation (Was: Re: making a checkerboard array?)

Posted by [Allan Whiteford](#) on Wed, 26 Sep 2007 16:51:06 GMT

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

---

Fiç½LDY Lajos wrote:

>

<snip>

>

> In C/C++/Fortran/... there is a rule that a memory cell can be changed  
> at most once in a statement, otherwise the result is undefined (and so  
> different implementations can create different results without violating  
> the standard). I guess a similar rule is in effect in IDL, too.

>

That's part of the question... does such a rule exist for IDL? You can change the same variable twice in Perl (the implementation is the standard etc. etc.) even with strict and warnings on it doesn't complain about something like:

```
$c=3;
$d[$c++]=$c++;
```

I've never found anything in the IDL manual which explicitly says either way what is allowed in IDL. Maybe I should write to the nice support people and ask if they have an opinion.

```
> These examples give syntax error prior IDL 6.0:
>
> IDL> print, !version
> { sparc sunos unix 5.3 Nov 11 1999}
> IDL> ((a=fltarr(3)))[1]=10 & print, a
> % Expression must be named variable in this context: <FLOAT   Array[3]>.
> % Execution halted at: $MAIN$
> IDL>
>
```

In IDL 6 they changed it so that setting a variable returns the variable set rather than the value of the variable set (i.e. an expression):

```
IDL6> help,(a=fltarr(3))
A          FLOAT   = Array[3]
```

```
IDL5> help,(a=fltarr(3))
<Expression>  FLOAT   = Array[3]
```

I wonder if this means that they think it's ok for things like ((a=fltarr(3)))[1]=10 to be written or if they changed it for some other purpose. Again, maybe a question for the support people.

Thanks,

Allan

```
>
> regards,
> lajos
>
>
>
```

---