

---

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

---