Kenneth P. Bowman wrote:
> In article <MPG.249c6dccba4246d098a6e1@news.giganews.com>,
> David Fanning <news@dfanning.com> wrote:
>
>> Folks,
>>
>> If you've already had a beer, you may want to work
>> on this one. Should we be concerned? Or is this just
>> one more example for the Sky is Falling article?
>> I've voiced my opinion, but it didn't satisfy the
>> complainant. What say you?
>>
>> curly=[-1.912,-1.852,-1.386,-0.818,-0.343,-0.343,-1.233]
>> larry=[-1.912,-1.852,-1.386,-0.818,-0.343,-0.343,-1.233]
>> curly = curly*!PI/180.
>> larry *= !PI/180.
>> diff=curly-larry
>> print,diff
>>
>> Why aren't the first set of differences (where the assignment
>> is made with the *= operator) *exactly* zero?
>>
>> Cheers,
>>
>> David
>
> My guess is that the compiler/interpreter does the multiplication and
> division operations in a different order in the two statements in the
> first case. The other two cases are syntactically identical, so the
> operations should be carried out in the same order.

Yes, adding some parentheses makes the results agree:

IDL> curly=[-1.912,-1.852,-1.386,-0.818,-0.343,-0.343,-1.233]
IDL> larry=[-1.912,-1.852,-1.386,-0.818,-0.343,-0.343,-1.233]
IDL> curly = curly*(!PI/180.)
IDL> larry *= !PI/180.
IDL> diff=curly-larry
IDL> print,diff
      0.00000   0.00000   0.00000   0.00000   0.00000
0.00000   0.00000

> *Why* the operations are done in a different order is the meta-
> question.

Well, in

    curly = curly * !pi / 180.

operations are just done left to right since they all have the same precedence. In

    larry *= some_value

some_value must first be calculated in order to use the *= operator.

Mike
--
www.michaelgalloy.com
Associate Research Scientist
Tech-X Corporation