

Re: Reason for operator precedence

Source: <http://sci.tech-archive.net/Archive/sci.math/2006-03/msg02419.html>

- *From:* briggs@xxxxxxxxxxxxxxxxxxxx
 - *Date:* 14 Mar 2006 09:00:19 -0600
-

In article <1142344511.262841.322440@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx>, matt271829-news@xxxxxxxxxxxx writes:

bri...@xxxxxxxxxxxxxxxxxxxx wrote:

In article
<1142342196.542632.294210@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx>, matt271829-news@xxxxxxxxxxxx writes:

Tony wrote:

Hi all.

Hope this isn't a silly question.

I was wondering what the reason is for having multiple levels of operator precedence?

Phrased another way, why is it that we don't just evaluate everything from left to right?

Having multiple levels of precedence obviously adds complexity, so I assume there must be some payback. However, I don't see what it is.

As far as addition/subtraction vs multiplication/division is concerned, one reason is to ensure that the distributive property of multiplication works sensibly. For example, we want $3*(4 + 6) = 3*4 + 3*6 = 3*(6 + 4) = 3*6 + 3*4$.

Re: Reason for operator precedence

Remember that what we're talking about here is merely a notational convention. It has nothing whatsoever to do with the distributive property of multiplication over addition.

You can express the distributive law for multiplication over division using parentheses:

$$a*(b+c) = (a*b) + (b*c)$$

Obviously you can. I meant to make it work without needing parentheses, but it seems that wasn't clear.

Ok. Try doing it using infix notation and the operator precedence convention of your choice. Remember your rule: no parentheses

Left to right doesn't work.

$b+c*a = a*b...$ and we're stuck

Right to left doesn't work.

$b+c*a = ...b*c$ and we're stuck.

Multiplication has precedence over addition doesn't work.

$a*...$ and we're stuck

Addition has precedence over multiplication doesn't work.

$a*b+c = a*b+...$ and we're stuck

Accordingly, trying to point to this case as a motivation for some particular choice of operator precedence seems ill conceived.

According to your argument, it follows that we are all using either Polish (prefix) or Reverse Polish (postfix) notation.

.