

Re: perl flawed or my fault

Source: <http://coding.derkeiler.com/Archive/Perl/comp.lang.perl.misc/2006-10/msg00345.html>

- *From:* anno4000@xxxxxxxxxxxxxxxxxxxxxxxxxxxxx
 - *Date:* 3 Oct 2006 18:36:55 GMT
-

Eric Schwartz <emschwar@xxxxxxxx> wrote in comp.lang.perl.misc:

"John W. Krahn" <someone@xxxxxxxx> writes:

```
defined($something) ? ($note_whom = $something) : ($otherthing =
'singapore');
```

Or better yet, write it out:

```
if (defined($something)) {
    $note_whome = $something;
} else {
    $otherthing = 'singapore';
}
```

I always recommend against the ?: operator unless it's for something small and self-contained. I avoid using it except for rvalues, unless not using it will make things unacceptably ugly. Yes, this is a vague standard. Here' an exampe of what is acceptable:

```
my $foo = $bar ? 0 : 1;
```

Here's something that isn't:

```
($var == 'foo' and $bar < 6 or $x > $y) ? ($this = something($else,
'maybe') or die "um, I dunno") : call_cows($potatoes);
```

One point about ternary ?: is that it is an expression that has a value, not primarily a means of flow control. The value should be the main point if the construct is used, not the side-effects of the code in the branches. In my view, small and self-contained enters the image only in so far as it can be hard to find a readable format for arbitrarily nested ?: with big complex branches. If such a format can be found, there is no limit to the complexity.

Re: perl flawed or my fault

I am of a split mind about using ?: in lvalue context. It *is* kind of obscure, but there are situations that simply call for it. One is the loop of a binary search, for example

```
while ( $low < $high - 1 ) {  
  my $mid = ($low + $high)/2;  
  $list->[ $mid ] le $targ ? $low : $high = $mid;  
}
```

The if-else equivalent to ?: is four lines, even cuddling the else. I can never bring myself to use it.

Anno

.