ParenParams (Ruby)

Problem

Precendence rules sometimes make it difficult to understand why a method call gives the answer it does. For example, the following came up in RubyTalk:12899

```
    p (2.4/0.2).to_i   # -> 12.0
    p ((2.4/0.2).to_i) #-> 11
```

A poster wanted to know why (x) != x.

Of course what's really happening is that Ruby is treating the parenthesis after the 'p' as the start of a parameter list. The matching closing parenthesis then closes the parameter list, and anything that follows it gets applied to the result of the function. Thus the above is the same as:

```
     p(2.4/0.2)  .to_i  # -> 12.0
     p((2.4/0.2).to_i)  #-> 11
```

Style Tip

If you have a method parameter that involves any degree of complexity or that starts with a parenthesized expression, put the whole parameter list in parentheses.

Discussion

This can be a cause of some pretty subtle bugs, so it's worth getting into the habit of using parenthesized parameter lists. Apart from a few common idioms (such as `require "name"`, and `attr :name`) where parentheses would seem strange, it's rarely a mistake to add them.

See also

- ButtParams

Contributors

PragDave

Part of the RubyStyleGuide.

Rev 6, Last edited at January 10, 2007 00:00 am by syarus / 68.95.131.252 (diff)
Approved by JimWeirich at January 09, 2007 21:23 pm