Ruby 1.8 - Bug #1979

parser confused by local variable assignment

08/22/2009 10:38 AM - coatl (caleb clausen)

Status:	Assigned
Priority:	Normal
Assignee:	matz (Yukihiro Matsumoto)
Target version:	
ruby -v:	1.8 and up

Description

=begin
I posted this before as a followup to bug #1801. Since there was no response and it seems to have been fixed independently of #1801 in the distant past, I'm filing it as a separate bug.

I can't decide if this is a different manifestation of bug #1801 or just a highly similar bug, maybe someone can enlighten me:

p = p m %(2)

I think the %(2) should be treated as a string, since it is in all other cases I know of (unless m is an lvar, which obviously it isn't here). However, it gets parsed as an operator and a parenthesized numeric literal.

In this variation of the expression, %(2) is a string, as I would expect:

p m %(2)

Is this the same as #1801?

This expression works as I expected for versions 1.4.0 thru 1.7.1. All versions after 1.7.1 don't work the way I expect. In versions before 1.4.0 it causes a syntax error. Jruby has identical behavior as recent mri rubies. In rubinius, it causes a stack dump.

Something interesting happened around about 1.7.1.
=end

History

#1 - 08/22/2009 03:18 PM - brixen (Brian Shirai)

=begin
Hi,

On Fri, Aug 21, 2009 at 6:39 PM, caleb clausenredmine@ruby-lang.org wrote:

> Bug #1979: parser confused by local variable assignment
> http://redmine.ruby-lang.org/issues/show/1979
>
> Author: caleb clausen
> Status: Open, Priority: Normal
> ruby -v: 1.8 and up
>
> I posted this before as a followup to bug #1801. Since there was no response and it seems to have been fixed independently of #1801 in the distant past, I'm filing it as a separate bug.
>
> I can't decide if this is a different manifestation of bug #1801 or just a highly similar bug, maybe someone can enlighten me:
>
> p = p m %(2)
>
> I think the %(2) should be treated as a string, since it is in all other cases I know of (unless m is an lvar, which obviously it isn't here). However, it gets parsed as an operator and a parenthesized numeric literal.
>
> In this variation of the expression, %(2) is a string, as I would expect:
>
> p m %(2)
>
> Is this the same as #1801?
>
> This expression works as I expected for versions 1.4.0 thru 1.7.1. All versions after 1.7.1 don't work the way I expect. In versions before 1.4.0 it

> causes a syntax error. Jruby has identical behavior as recent mri rubies. In rubinius, it causes a stack dump.

Could you elaborate on what you get with Rubinius? I get 'nil' for the first example (assuming I'm typing this in right) and NoMethodError for 'm' in the second. Could you give a screen dump or something?

Thanks,
Brian

> Something interesting happened around about 1.7.1.

> http://redmine.ruby-lang.org

=end

#2 - 10/28/2009 02:42 AM - naruse (Yui NARUSE)

- Category set to core

- Status changed from Open to Assigned

- Assignee set to matz (Yukihiro Matsumoto)

=begin

=end