

Ruby master - Feature #6287

nested method should only be visible by nesting/enclosing method

04/13/2012 04:55 PM - botp (bot pena)

Status:	Closed	
Priority:	Normal	
Assignee:	matz (Yukihiro Matsumoto)	
Target version:	3.0	
Description		
<pre>def test1 def test2 p "i am test2" end test2 end test1 "i am test2" # ok test2 "i am test2" # not ok (imho), this should fail; ie test2 should only be visible by test1, not by outside.</pre>		
Related issues:		
Is duplicate of Ruby master - Feature #4085: Refinements and nested methods		Closed 11/24/2010

History

#1 - 04/13/2012 05:34 PM - shyouhei (Shyouhei Urabe)

- Description updated

I know what you mean, but I think this is a limitation of Ruby's OOP nature. In Ruby, everything belongs to an object. So a method must belong to an object, not another method.

#2 - 04/13/2012 10:58 PM - mame (Yusuke Endoh)

- Description updated

- Status changed from Open to Assigned

- Assignee set to matz (Yukihiro Matsumoto)

- Target version changed from 2.0.0 to 3.0

Hello,

There is a similar proposal [#4085](#) based on more general idea.

Below is my personal opinion.

Ruby's def statement is not a static definition, but a dynamic execution that defines a method in the context. This design allows you to define a method that is defined in some condition:

```
if windows?
  def foo
  end
end
```

, singleton methods for each element in an array:

```
ary.each do |elem|
  def elem.foo
```

```
end
end
```

, and so on.

In the same manner, a "nested" method definition you said:

```
def define_foo
  def foo
  end
end
```

should define a method named "define_foo" that defines a method named "foo." I think it is consistent.

--

Yusuke Endoh <mame@tsg.ne.jp>

#3 - 04/14/2012 12:59 AM - cout (Paul Brannan)

On Fri, 2012-04-13 at 22:58 +0900, mame (Yusuke Endoh) wrote:

In the same manner, a "nested" method definition you said:

```
def define_foo
  def foo
  end
end
```

should define a method named "define_foo" that defines a method named "foo." I think it is consistent.

I agree. The behavior is consistent. But I'm yet to find a case where this is useful.

Paul

#4 - 01/07/2013 12:28 PM - shugo (Shugo Maeda)

- *Status changed from Assigned to Closed*

I've implemented all features requested by Matz, so I close this ticket.

The current spec is described at <https://bugs.ruby-lang.org/projects/ruby-trunk/wiki/RefinementsSpec>.

If you have any request for Refinements, please file a new ticket.

However, the feature set of Ruby 2.0.0 has already been frozen, so the spec won't be changed in Ruby 2.0.0 unless otherwise permitted by Matz or Endoh-san.