Ruby master - Feature #7019
allow `private` and `protected` keywords to take blocks
09/13/2012 04:02 PM - alexeymuranov (Alexey Muranov)

Status: Rejected
Priority: Normal
Assignee: matz (Yukihiro Matsumoto)
Target version: 3.0

Description
I like to indent my private methods one level deeper, but this indentation is inconsistent with the syntax:

class C
def f
  #
end
private
def g
  #
end
end

I think it would be nice if the private keyword could take a block, then i would write:

class C
def f
  #
end
private do
def g
  #
end
end
end

Related issues:
Has duplicate Ruby master - Feature #8497: private, protected, private_cons... Rejected
Has duplicate Ruby master - Feature #16276: For consideration: "private do..... Open

History
#1 - 09/16/2012 10:57 PM - trans (Thomas Sawyer)
I've often wondered about being able to use:

private def f
  #
end

I think it's true that the public/private/protected "modifiers" have always felt a bit awkward --which accounts for the variety of indentation and comment "highlights" that have been used with them.

#2 - 09/17/2012 10:13 PM - shyouhei (Shyouhei Urabe)
I know you like blocks but this proposal doesn't make sense to me because public/private/protected are declarations, not something done. So it's odd to say "private DO something". I'd rather +1 to private def, which is natural (to me at least).

#3 - 09/17/2012 11:24 PM - alexeymuranov (Alexey Muranov)
What about

privately do
end

? ;)
(I understand that a good reason is needed to add a new keyword).
Here is a use case I have made up. The following does not work now:

class C
  def f
    def g
      #
      end
    private
    def h
      #
    end
  end
end

c = C.new
f # => NameError: undefined local variable or method 'private' for ...

So a block syntax could make sense here:

class C
  def f
    def g
      #
    end
    privately do
      def h
        #
      end
    end
  end
end

I don't know why you would expect this to work, private, public, etc. are methods on Module and your context is not a subclass of Module.

Why should we pollute every object with a new methods privately, publicly and protectedly (which isn't a word)?

+1 for "private def foo". I have never seen a good justification for why visibility is better as a value on the current frame, nor for why it's better to set a method's visibility after declaration rather than along with the declaration.

The fact that visibility lives on the frame also means implementations that attempt to optimize frames away (like JRuby does and eventually MRI will want to do) have a harder time of it.

So two changes:

- def will require to return the method defined instead of nil
- private will require to accept a Method and not just a Symbol

I believe these two changes where already requested (can't find the bug reports yet).

This approach helps also to describe methods beyond documentation markers, making it clear when a method is private inline with the method definition and not after the method has been defined.

I like this.

Would need to accept array of those too (whether its a method or symbol) for things like:
Excuse me Thomas, what's the point in a "private attribute accessor"?

Probably not much, but it is possible. Could be "protected" instead as well. Also, other helpers can be created besides attr. My point is only that it needs to accept an array should the helper return multiple methods/symbols.

Or are you thinking that private def would be some kind of keyword thing?

mame-san, could you judge this ticket?

IMO, it is too slow to introduce such a big feature.

IMO, it is too slow to introduce such a big feature.

Completely agreed.

Please do not forget also module_function and such. (Or are they different in some sense?).

Alexey.

#13 - 10/27/2012 08:54 PM - alexeymuranov (Alexey Muranov)
Please do not forget also module_function and such. (Or are they different in some sense?).

Alexey.

#14 - 01/20/2015 06:22 AM - nobu (Nobuyoshi Nakada)
- Description updated

#15 - 01/22/2018 12:27 AM - kou (Kouhei Sutou)
- Status changed from Assigned to Rejected

#3753 adds private def ... support. If you still want this feature, please reopen this issue.

#16 - 10/24/2019 01:52 AM - shyouhei (Shyouhei Urabe)
- Has duplicate Feature #16276: For consideration: "private do...end" / "protected do...end" added