Would it be a good idea to compress these 4 methods to 2 methods?

public - Can set both instance and class methods to public by passing them in as symbol
private - Can set both instance and class methods to private by passing them in as symbol

and enable them to be called at top of class? Is this a good idea? It would clean up Module# and encourage use of these two as methods rather than keywords

History

#1 - 03/26/2013 07:49 AM - drbrain (Eric Hodel)
- Category set to core

=begin
If private_class_method is removed, how will the class method foo be set to private here:

class C
def self.foo() end
def foo() end
private :foo
end
=end

#2 - 03/27/2013 02:09 AM - EdwardCh (Edward Chen)
"Stuff" the private() method to take the explicit options. A possible solution would be:
private {:instance => [], :class => [foo]}

In this way, although it adds a bit of complexity to the private and public methods, we would have reduced the method count of Module by 2 and simplified a class view => I can see all the public, private methods at a glance.

#3 - 03/27/2013 02:30 AM - drbrain (Eric Hodel)
- Status changed from Open to Assigned
- Assignee set to matz (Yukihiro Matsumoto)
- Target version set to 3.0

#4 - 03/27/2013 06:10 AM - nobu (Nobuyoshi Nakada)
Class methods and instance methods are different things. I don't see advantages of this proposal.