

Ruby - Feature #8273

Make it possible to treat objects as falsy in order to make NilObjects more convenient

04/16/2013 08:11 AM - prijutme4ty (Ilya Vorontsov)

Status:	Rejected	
Priority:	Normal	
Assignee:		
Target version:		
Description <p>NilObject is quite a useful pattern. But refactoring from conditionals to null-object can be very painful because an instance of NilObject will be treated as truthy so code ... if obj can change its behavior. It'd be great to treat such objects as falsy. Most simple and consistent solution is to make method like #to_boolean or may be #__to_bool (not to break compatibility) which returns false for NilClass and FalseClass and true otherwise unless redefined.</p> <p>I don't know whether performance impact'll be high or no.</p> <p>It's one of a few things which cannot be reimplemented using ruby itself, so there is no possibility to construct a gem as a PoC</p>		
Related issues: <p>Is duplicate of Ruby - Feature #6180: to_b for converting objects to a boolea... Rejected 03/21/2012</p>		

History

#1 - 04/16/2013 01:45 PM - headius (Charles Nutter)

I think you might have submitted this before, but I'd like to enter my *strong* opposition to this.

- It most definitely would affect performance. Any language construct that needs to check for truthiness would suddenly have to do a method call rather than simply an identity check.
- I think it is a terrible thing that other languages like Python and PHP make it possible for the same object to be true one moment and false the next (e.g. a list that empties out or a string that is blanked). What is true should always be true; what is false should always be false.

The last point is my really strong position. I do not think it should ever be possible for the contents of an object to define its truthiness, and by extension for changes to the contents of an object to change its truthiness.

#2 - 04/16/2013 09:07 PM - rosenfeld (Rodrigo Rosenfeld Rosas)

I should agree with Charles although I understand Ilya's concerns. Maybe another attempt to get this implemented would be simply requesting native support for a NilObject class in such a way that falsey would be either false, nil or an instance of NilObject. What do you think about such proposal?

#3 - 04/16/2013 11:51 PM - headius (Charles Nutter)

@rosenfeld That would certainly be less impact, but not zero-impact. I do not have a strong opinion one way or the other on that sort of change, but I still suspect it's unlikely to be approved. Changing the semantics of truthiness is a pretty major thing.

#4 - 04/17/2013 03:50 AM - prijutme4ty (Ilya Vorontsov)

[@headius \(Charles Nutter\)](#)

I understand your oppositions. IMO, NilObject should be a kind of immutable/singleton object so that object couldn't change its truthiness unless method is redefined. (but in ruby there're already many ways to shoot own leg)

@rosenfeld

It sounds reasonable. But it makes impossible to make EmptyList as a subclass of List.

#5 - 04/17/2013 10:04 AM - matz (Yukihiro Matsumoto)

- Status changed from Open to Rejected

Rejected for the same reason as [#6180](#).

Matz.

#6 - 12/23/2021 11:40 PM - hsbt (Hiroshi SHIBATA)

- Project changed from 14 to Ruby