

Ruby master - Bug #10795

to_s returns references to self if called on string

01/28/2015 03:58 PM - fra.boffa (Francesco Boffa)

Status: Rejected	
Priority: Normal	
Assignee:	
Target version:	
ruby -v: ruby 2.1.5p273 (2014-11-13 revision 48405) [x86_64-darwin13.0]	Backport: 2.0.0: UNKNOWN, 2.1: UNKNOWN, 2.2: UNKNOWN
Description	
<p>This is not actually a bug, but rather a strong violation of the Principle of least astonishment.</p> <p>Lately I found a bug in the Rails project I work to. There was a method concatenating two float values with a comma, converting them to string like this:</p> <pre>def coords_to_string latitude.to_s << ", " << longitude.to_s end</pre> <p>The bug happened when we passed latitude and longitude as strings instead of floats.</p> <p>Actually, 5.12345.to_s returns a new string object and any operation made on that string wont affect the original 5.12345. Instead "5.12345".to_s will return the same instance of that string, so that the << method will affect the original string.</p> <p>So we found that our latitude variable was growing on each call and after a while it become "5.12345,13.12345,13.12345,13.12345,13.12345,13.12345"</p> <p>This was probably made like this for performance reasons, however we found it a clear violation of the POLA, whereas we expected to_s to always return a new instance of that string.</p> <p>Hope, this may help. Francesco Boffa</p>	

History

#1 - 01/28/2015 09:08 PM - marcandre (Marc-Andre Lafortune)

- Status changed from Open to Rejected

Remember, the principle of least astonishment applies to Matz only :-)

Changing this would, among other things, be a source of incompatibilities.

In any case, your code is better written

```
def coords_to_string
  "#{latitude},#{longitude}"
end
```

#2 - 01/28/2015 10:45 PM - sawa (Tsuyoshi Sawada)

Marc-Andre Lafortune wrote:

In any case, your code is better written

```
def coords_to_string
  "#{latitude},#{longitude}"
end
```

or

```
def coords_to_string
  [latitude, longitude].join(",")
end
```

end

#3 - 01/29/2015 08:26 AM - fra.boffa (Francesco Boffa)

For sure, our code was not the ideal for that task, and indeed, I already had changed it to one of your sane alternatives and the original committer has been appropriately thrown by the window :).

I insist, however, that this should at least be made clearer in the documentation of all the `to_s` methods available in the standard library.

Thanks for your attention.

#4 - 01/29/2015 03:58 PM - marcandre (Marc-Andre Lafortune)

Francesco Boffa wrote:

I insist, however, that this should at least be made clearer in the documentation of all the `to_s` methods available in the standard library.

Not sure what you mean by "all" `to_s` methods. Only `String#to_s` returns the receiver.

The document is quite explicit, both in the interface and the description. OTOH, it didn't mention the effect for subclasses, so I modified it to follow that of `Array#to_a`.