

Ruby master - Feature #12059

`Array#single?`, `Hash#single?`

02/09/2016 03:37 PM - sawa (Tsuyoshi Sawada)

Status:	Open
Priority:	Normal
Assignee:	
Target version:	
Description	
There are some use cases when one wants to check if an array or a hash has exactly one element. I propose <code>Array#single?</code> and <code>Hash#single?</code> that checks for such cases and returns either true or false. This is an analogy from the <code>empty?</code> method on the respective class.	
<ul style="list-style-type: none">When creating an inflectional form out of an array:	
<pre>a = ["object1", "object2"] "There #{a.single ? "is" : "are"} #{a.length} #{a.single? ? "object" : "objects"}." # => "There are 2 objects."</pre>	
<ul style="list-style-type: none">When checking if all elements of the array are the same:	
<pre>[1, 2, 2, 1].uniq.single? # => false [1, 1, 1, 1].uniq.single? # => true</pre>	

History

#1 - 02/09/2016 03:41 PM - sikachu (Prem Sichanugrist)

I feel like the usage of this method wouldn't be generic enough, and could be accomplished easily by `.size.one?`

#2 - 02/09/2016 04:31 PM - mame (Yusuke Endoh)

I'm neutral for the proposal itself. If it is accepted, I think "singleton?" is the best name.

In mathematics, a singleton ... is a set with exactly one element.

[https://en.wikipedia.org/wiki/Singleton_\(mathematics\)](https://en.wikipedia.org/wiki/Singleton_(mathematics))

--

Yusuke Endoh mame@ruby-lang.org

#3 - 02/09/2016 09:51 PM - danielpclark (Daniel P. Clark)

Ruby currently supports the `one?` method on both Hash and Array.

```
[1].one?
# => true
{a: 1}.one?
# => true
[1].method(:one?).owner
# => Enumerable
```

#4 - 02/10/2016 02:11 AM - nobu (Nobuyoshi Nakada)

`[1, false, nil].one?` also returns true.

From ri `Enumerable#one?`:

```
enum.one? [{ |obj| block }] -> true or false
```

Passes each element of the collection to the given block. The method returns true if the block returns true exactly once. If the block is not given, `one?` will return true only if exactly one of the collection members is true.

```
%w{ant bear cat}.one? { |word| word.length == 4 } #=> true
```

```
%w{ant bear cat}.one? { |word| word.length > 4 } #=> false
%w{ant bear cat}.one? { |word| word.length < 4 } #=> false
[ nil, true, 99 ].one? #=> false
[ nil, true, false ].one? #=> true
```

#5 - 02/10/2016 04:02 AM - danielpclark (Daniel P. Clark)

Right. Given the original examples by the OP Enumerable#one? works.

```
a = ["object1", "object2"]
"There #{a.one? ? "is" : "are"} #{a.length} #{a.one? ? "object" : "objects"}."
# => "There are 2 objects."
[1, 2, 2, 1].uniq.one?
# => false
[1, 1, 1, 1].uniq.one?
# => true
```