

## Ruby master - Feature #11323

### Documentation update on how uniq works / guarantee of order

07/02/2015 01:39 AM - wilburlo (Daniel Lo)

<b>Status:</b>	Open
<b>Priority:</b>	Normal
<b>Assignee:</b>	
<b>Target version:</b>	
<b>Description</b>	
Greetings,	
I was looking at Array.uniq and I was wondering how the code made the array unique.	
There are 2 different possible outcomes for making an array unique.	
For example:	
[1,2,1]	
The first value is kept and all subsequent duplicate values are removed: [1,2]	
or	
The array is made unique, order is not retained: [2,1]	
Would the ruby team consider adding a guarantee of order (first seen/first kept) is adding this to the Array.uniq specification? This is what happens in practice (irb), having this as part of the specification would be nice.	
I looked at the code <a href="http://ruby-doc.org/core-2.2.1/Array.html#method-i-uniq">http://ruby-doc.org/core-2.2.1/Array.html#method-i-uniq</a> however, I wasn't able to determine exactly how it worked. :(	
Thank you!	
Regards,	
Daniel Lo	

### History

#### #1 - 07/02/2015 03:34 AM - 0x0dea (D.E. Akers)

Array#uniq is implemented using a hash table; since hashes in Ruby preserve insertion order, so too does Array#uniq. The following simplified translation to Ruby should help clarify what's going on in rb\_ary\_uniq():

```
class Array
  def uniq
    hash = {}
    each { |val| hash[val] = val }
    hash.values
  end
end
```

There is precedent in Array's documentation for mentioning that a method is order-preserving, and it seems that #uniq should certainly do so.