Ruby trunk - Misc #15224

[DOCs] Minor inconsistency in class Array #initialize_copy - https://ruby-doc.org/core-2.5.1/Array.html#method-i-initialize_copy

10/13/2018 10:18 AM - shevegen (Robert A. Heiler)

Status:	Open	
Priority:	Normal	
Assignee:		

Description

Today I looked at:

https://ruby-doc.org/core-2.5.1/Array.html#method-i-initialize_copy

The example to this method is this:

```
a = [ "a", "b", "c", "d", "e" ]
a.replace([ "x", "y", "z" ])   #=> ["x", "y", "z"]
a                              #=> ["x", "y", "z"]
```

What confused me was that I was looking at the method called initialize_copy
but the example showed .replace().

I then looked at #replace there:

https://ruby-doc.org/core-2.5.1/Array.html#method-i-replace

And it was virtually identical to initialize_copy.

I assume the examples for .replace() are correct; and perhaps initialize_copy
is just an alias? I am not sure, but I would like to suggest to make the documentation,
in particular the example, a bit more consistent.

When you click on "view source" to look at the C code, they show the very same
content, so I believe that initialize_copy is merely an alias to replace; but I tried
this and they are not fully equivalent:

```
x = [1,2,3] # => [1, 2, 3]
x.initialize_copy [4,5,6]

Traceback (most recent call last):
        2: from /System/Index/bin/irb:11:in `<main>'
        1: from (irb):2

NoMethodError (private method `initialize_copy' called for [1, 2, 3]:Array)
```

Yet:

```
x.replace [4,5,6] # => [4, 5, 6]
```

works. So I assume that initialize_copy is like .replace() but is a private
method instead.

Perhaps it may help to add a sentence below the documentation of
replace(), to explain what the use case for initialize_copy is. Or to perhaps
mention that it is an alias.

At the least how it is right now is that people may read initialize_copy,
but then see an example of #replace. (Perhaps an example for
initialize_copy may help, but either way, I think the current docu-example
is not ideal).

History

#1 - 10/13/2018 02:25 PM - nobu (Nobuyoshi Nakada)

- Description updated

Yes, Array#initialize_copy and Array#replace share the same implementation (and the doc of course).
The only difference is that initialize, initialize_copy, initialize_clone, initialize_dup and respond_to_missing? are defined as private automatically.
What you should know about initialize_copy to use it, is that you should never use it directly.
Should we remove the name in the doc?

#2 - 10/13/2018 02:26 PM - nobu (Nobuyoshi Nakada)

- Description updated