

## Ruby master - Feature #5644

### add Enumerable#exclude? antonym

11/17/2011 09:38 AM - sunaku (Suraj Kurapati)

<b>Status:</b>	Feedback
<b>Priority:</b>	Normal
<b>Assignee:</b>	matz (Yukihiro Matsumoto)
<b>Target version:</b>	
<b>Description</b>	
Please add Enumerable#exclude? as antonym of Enumerable#include?	
This allows me to construct Boolean expressions more pleasantly:	
if File.exist? some_file and not some_list.include? some_file	
Can be written as:	
if File.exist? some_file and some_list.exclude? some_file	
Thanks for your consideration.	

#### History

##### #1 - 11/17/2011 10:09 AM - trans (Thomas Sawyer)

Hey, a use for functors!

```
module Kernel
  def not
    Functor.new do |op,*a,&b|
      !send(op,*a,&b)
    end
  end
end
```

some\_list.not.include? some\_file

Nothing against Enumerable#exclude? though. Seems reasonable.

##### #2 - 03/28/2012 12:28 AM - mame (Yusuke Endoh)

- Status changed from Open to Assigned

- Assignee set to matz (Yukihiro Matsumoto)

##### #3 - 03/28/2012 01:25 AM - matz (Yukihiro Matsumoto)

- Status changed from Assigned to Feedback

from that logic, don't we need to add antonyms to every predicate methods?  
could you show us why include? is special?

Matz.

##### #4 - 03/29/2012 08:26 AM - sunaku (Suraj Kurapati)

Hi Matz,

I didn't ask for antonyms for all predicates; only for #exclude?.

The reason for #exclude? is for more "natural" boolean expressions:

```
if File.exist? some_file and some_list.exclude? some_file
if File.exist? some_file and not some_list.include? some_file
```

That "not SOMETHING.include? SOMETHING" pattern appears often in my code, so that's why I created this request. Of course, me saying "appears often" does not constitute as solid evidence in support of adding #exclude? so I have no choice but to accept your judgement on this request. It's your call.

Thanks for your consideration.

**#5 - 03/29/2012 12:41 PM - trans (Thomas Sawyer)**

I'll throw my hat in with #exclude? too. There's been a number of times that I would have liked to have it. "not include?" is a rather common predication and it's nice when code can line up neatly.

**#6 - 03/29/2012 02:29 PM - now (Nikolai Weibull)**

On Thu, Mar 29, 2012 at 01:26, sunaku (Suraj Kurapati) [sunaku@gmail.com](mailto:sunaku@gmail.com) wrote:

Issue [#5644](#) has been updated by sunaku (Suraj Kurapati).

The reason for #exclude? is for more "natural" boolean expressions

I don't think #exclude? really conveys what's being done very well. Yes, "exclude" is the antonym of "include", but the meaning of %[a b c].include? x is very natural, whereas %[a b c].exclude? x isn't. Does it mean that %[a b c] contains the elements that should be excluded and is x among them, or is x not included among the elements of %[a b c]?

**#7 - 03/30/2012 11:02 PM - matz (Yukihiko Matsumoto)**

OK, you think negative for include? is special. Understood. But as Nikolai pointed out, exclude? is not the best name for the function. Any alternative?

Matz.

**#8 - 03/30/2012 11:09 PM - rosenfeld (Rodrigo Rosenfeld Rosas)**

At first I agreed with Nikolay, but then I changed my minded because the method is called "exclude?" with a question mark, not "exclude", so I don't think anyone would expect that it would actually remove some element.

**#9 - 03/30/2012 11:13 PM - rosenfeld (Rodrigo Rosenfeld Rosas)**

I think I've misunderstood the question posed by Nikolai. I've just read it again but I think that the other meaning presented by him doesn't make any sense. "does the array contain the elements that should be excluded?". Really? I read this like in English:

Does [1, 3, 5] exclude 4?

Ask someone that doesn't know anything about programming and see what will she answer to this question.

**#10 - 03/30/2012 11:18 PM - rosenfeld (Rodrigo Rosenfeld Rosas)**

The most common antonym is "exclude", but maybe we could use "omit" if you prefer:

<http://www.synonym.com/antonym/include/>

**#11 - 03/30/2012 11:49 PM - trans (Thomas Sawyer)**

There really is no better term b/c all such terms are going to have the same connotations.

As with "include" if you add an "s" to the word then it reads more like typical English, i.e. "a excludes b ?". To use the singular form you have to add a modal verb like "does a exclude b ?" Which makes it easy to see that this is the right meaning.

**#12 - 03/31/2012 12:59 AM - now (Nikolai Weibull)**

On Fri, Mar 30, 2012 at 16:02, matz (Yukihiko Matsumoto) [matz@ruby-lang.org](mailto:matz@ruby-lang.org) wrote:

OK, you think negative for include? is special. Understood. But as Nikolai pointed out, exclude? is not the best name for the function. Any alternative?

How about changing the definition of Enumerable#none?, #any?, and #all? to take an optional argument that is compared against each element using #==. Then we have

if File.exist? some\_file and some\_list.none? some\_file

I would not write it like that – I'd still use not a.include? b, which

I think is hard to beat – but then we at least don't need to come up with a new name and these three methods gain semantics that I've felt they were lacking.

**#13 - 11/20/2012 10:21 PM - mame (Yusuke Endoh)**

- *Target version set to 2.6*

**#14 - 11/20/2012 10:33 PM - trans (Thomas Sawyer)**

Too bad we can't use symbols like €.

**#15 - 12/25/2017 06:15 PM - naruse (Yui NARUSE)**

- *Target version deleted (2.6)*