

Ruby trunk - Feature #15722

`Kernel#case?`

03/22/2019 07:12 AM - sawa (Tsuyoshi Sawada)

Status:	Open
Priority:	Normal
Assignee:	
Target version:	
Description	
<p>I often want to use === to match a single object on the right side against multiple objects on the left, as is used in case-constructions, just to return a truth value, and end up writing like this:</p>	
<pre>bar # => "bar" flag1 = case bar; when "foo", "bar", "baz"; true; end # => true flag2 = case bar; when Symbol, String; true; end # => true</pre>	
<p>I propose Kernel#case? that should work like this:</p>	
<pre>bar # => "bar" bar.case?("foo", "bar", "baz") # => true bar.case?("qux") # => false bar.case?(Symbol, String) # => true bar.case?(Array) # => false bar.case? # => false</pre>	
<p>It is similar to Rails' in?, but it differs from it in that it uses === for comparison, not ==.</p>	

History

#1 - 03/22/2019 09:13 AM - osyo (manga osyo)

How about Enumerable#case? ?

Comparing with === like Enumerable#include?.

```
bar # => "bar"
["foo", "bar", "baz"].case? bar # => true
["qux"].case? bar # => false
[Symbol, String].case? bar # => true
[Array].case? bar # => false
```

#2 - 03/22/2019 10:04 AM - sawa (Tsuyoshi Sawada)

osyo (manga osyo) wrote:

How about Enumerable#case? ?

Comparing with === like Enumerable#include?.

```
bar # => "bar"
["foo", "bar", "baz"].case? bar # => true
["qux"].case? bar # => false
[Symbol, String].case? bar # => true
[Array].case? bar # => false
```

What is to be evaluated is bar, not the objects that you put in the arrays. bar has to be the receiver. That also matches with how case-construction works. Furthermore, having an array as in your proposal requires additional array to be created, which will be immediately disposed.

#3 - 03/22/2019 04:25 PM - sawa (Tsuyoshi Sawada)

If such method, if any, should be rather defined on Object, that would be fine. I was not sure whether I should have requested this for Kernel or for Object.

#4 - 03/22/2019 05:11 PM - shevegen (Robert A. Heiler)

This is an interesting idea. I have not thought through all implications but not withholding that, I like it so +1 from me. I think matz has to decide whether this may be useful. Since I love case/when in ruby in general, any extension, even secondary ideas, seem useful to me. :)

Although I have to admit, I am not sure if I yet had a use case as suggested by sawa, but I still think it is a good idea.

I think whether it is on Kernel or Object or Enumerable is not the primary concern (that can be decided anyway) - if matz is too busy right now then perhaps this could be suggested for the next developer meeting to get more feedback.

I did not know about Rails in?, but I do not like the name "in?". I have no problem with "case?".

One possible question from the core team might be whether the use case is sufficient (for many ruby users). I can not answer that myself, to be honest, if that question would come up - but I still like the idea.

PS: I almost overlooked the comment by osyo; I agree, that is a different suggestion so I was confused for a moment. The original suggestion by sawa is clearer IMO and more appropriate (to the proposed suggestion); wording it like:

```
["foo", "bar", "baz"].case? bar
```

would be different, and is actually a bit confusing to me. So it should best be ignored; sawa's original description is clearer and makes more sense to me, but this is just an aside - I only was confused for a moment when I read the second code part by osyo.