

Ruby - Feature #11167

Allow an attr_ variant for query-methods that end with a question mark '?' character, such as: def foo? returning @foo

05/21/2015 07:05 PM - shevegen (Robert A. Heiler)

Status:	Rejected	
Priority:	Normal	
Assignee:		
Target version:		
Description		
<p>Hi guys,</p> <p>Hi nobu :)</p> <p>Also hi matz if matz reads this, and of course the rest of the core team and everyone else.</p> <p>Today on IRC, this mini-discussion happened (I show a snippet):</p> <pre><apeiros> I really miss attr_query or whatever you want to name it <apeiros> which would generate a ? method too <jhass> apeiros: crystal has :P getter? <apeiros> nice</pre> <p>Ok, so the language crystal has something ruby does not have.</p> <p>We can't let those newcomers get away with making ruby look old now can we!</p> <p>I use ruby not crystal but I very often use methods that end with a '?' query mark in ruby. It helps me in simple if clauses such as:</p> <pre>if hash.has_key? if hash.key? if cat.is_hungry?</pre> <p>(In the latter, it might be a cat of class Cat instance, with an instance variable called @is_hungry, and when the cat is fed with food, it is not hungry logically.)</p> <p>We can generate these @ivars through attr_* right now as is already, such as:</p> <pre>attr_reader :foo def foo; @foo; end attr_writer :foo def foo=(i); @foo = i; end attr_accessor :foo ^^^ Combines the above two methods into one.</pre> <p>But we have no way to designate methods that end via '?'.</p> <p>I do not know which API call would be nice. apeiros on IRC suggested attr_query</p> <p>I am fine with that. (The name is secondary for me, I would like to have this feature available - what name it would then have is not the main issue for me.)</p>		

apeiros then also suggested this syntax:

All `attr_*` that would end with a `?` token, would be a combination of `attr_reader` and also a variant of the above that has a `'?` token, so for example:

```
attr_reader :foo?
```

Would create both a method `foo()` and `foo?()`.

People who do not need this, can continue to use:

```
attr_reader :foo
```

just fine.

So perhaps this suggestion is even better than a new method (such as through `attr_query()`)

(I also have added one more line from apeiros, not sure if I understood it, but I think the above explanation should suffice - here is the other suggestion he did:)

```
apeiros> e.g. attr_reader :foo? -> foo? // attr_accessor :foo? -> foo= + foo? // all with @foo of course. and foo? returning true/false.
```

Ok, that's it.

Thanks for reading!

Related issues:

Related to Ruby - Feature #12046: Allow `attr_reader :foo?` to define instance ...

Rejected

Is duplicate of Ruby - Feature #10720: A proposal for something like: `attr_...`

Rejected

History

#1 - 05/21/2015 11:02 PM - nobu (Nobuyoshi Nakada)

- Is duplicate of Feature #10720: A proposal for something like: `attr_reader :foo?` - with the trailing `'?` question mark added

#2 - 05/21/2015 11:03 PM - nobu (Nobuyoshi Nakada)

- Description updated

#3 - 05/22/2015 12:43 PM - matz (Yukihiro Matsumoto)

Hi,

Is there any real-world use-case for the proposal?
It seems

```
def foo?  
  @foo  
end
```

is just fine.

Matz.

#4 - 05/22/2015 03:28 PM - djberg96 (Daniel Berger)

Abandoned all hope:

<http://blade.nagaokaut.ac.jp/cgi-bin/scat.rb/ruby/ruby-core/5796>

<https://www.ruby-forum.com/topic/135195>

#5 - 05/22/2015 05:44 PM - spatulasnout (B Kelly)

Hi Matz,

Yukihiro Matsumoto wrote:

Is there any real-world use-case for the proposal?

I've wanted this feature for years. :)

Here's an example from code I'm working with today:

```
def startup_complete?  
  @startup_complete  
end
```

I would have preferred:

```
attr_reader :startup_complete?
```

It seems

```
def foo?  
  @foo  
end
```

is just fine.

Perhaps fine, but clunky. :)

In other words, why do we ever use attr_reader (etc.) at all?

We use attr_* because it's preferable to writing a bunch of mini-functions.

The reasons are identical for the '?' variant.

Regards,

Bill

#6 - 05/23/2015 11:28 AM - duerst (Martin Dürst)

Daniel Berger wrote:

Abandoned all hope:

<http://blade.nagaokaut.ac.jp/cgi-bin/scat.rb/ruby/ruby-core/5796>

<https://www.ruby-forum.com/topic/135195>

I think that this shows that there are several people who really like to use a foo? getter with a foo= setter. What it doesn't show that there are enough such people that it would balance out the confusion for those who think that a foo= setter goes together with a foo getter, at least by default.

With the current state, the main case is covered, nobody gets confused, and those who want a foo?/foo= pair can easily get it either as Matz showed above (easily reduced to one line) or by redefining the attr_* methods; the later is also very easy as it is usually about the first example given when explaining metaprogramming.

#7 - 02/16/2016 06:54 AM - matz (Yukihiro Matsumoto)

- Status changed from Open to Rejected

See [#12046](#).

Matz.

#8 - 07/09/2019 12:08 AM - mame (Yusuke Endoh)

- Related to Feature #12046: Allow attr_reader :foo? to define instance method foo? for accessing @foo added

#9 - 01/10/2020 06:33 AM - anders (Anders Bälter)

Like that!

sudo (Sudo Nice) wrote:

How about implementing it similarly to Crystal?

```
attr_accessor? :foo
```

<https://bugs.ruby-lang.org/issues/5781#note-16>