

Ruby master - Feature #12096

New notation for instance variables and class variables

02/21/2016 03:09 PM - sawa (Tsuyoshi Sawada)

Status:	Closed
Priority:	Normal
Assignee:	
Target version:	
Description	
In order to create symbols that include non-word characters, we have the <code>'...'</code> notation:	
<pre>: 'foo-bar' : "foo-bar"</pre>	
What about extending this notation to instance variables and class variables? The use case is to solve the problems raised in #12046 , #11167 , #10720 . There, the problem was that, even though we can have methods like <code>foo?</code> or symbols like <code>:foo?</code> , we cannot have corresponding instance variables. My proposal is to introduce a similar notation for instance variables:	
<pre>@ 'foo?' @ "foo?"</pre>	
I can't think of a similar use case for class variables, but there is no reason to make class variables to behave different from instance variables more than necessary, so there can also be the corresponding notation for class variables:	
<pre>@@ 'foo?' @@ "foo?"</pre>	
Right now, these notations raise syntax errors, so I don't think it conflicts (raises ambiguity) with the existing syntax.	

History

#1 - 02/21/2016 04:20 PM - sawa (Tsuyoshi Sawada)

And of course, the range of available names for these variables should be extended. And `attr_reader :foo?` should also be allowed.

Another use case may be a shorthand for `instance_variable_set/instance_variable_get` with dynamically generated names when the receiver is omitted (self). Instead of:

```
instance_variable_set("@#{some_expression}", some_value)
```

we can do:

```
@#{some_expression} = some_value
```

#2 - 02/22/2016 11:41 PM - shevegen (Robert A. Heiler)

I like the idea to some extent but the syntax strikes me as somewhat peculiar.

```
@ 'foo?'
```

In ruby one has to look quite closely to spot some errors on a first glance; the `did-you-mean-gem` helps a bit but I think the human brain also has to process a lot of syntax. I think my problem with this here is the `@`. I myself do not really use `@@` but I use `@` a lot. But again, my reservation is not about the idea itself, more the specific syntax suggested.

I would prefer `@` to stay simple, without a lot of side effects or shortcuts.

And `attr_reader :foo?` should also be allowed.

Yes, I agree here, but I think `:foo?` may be easier to see too than the `@foo?`. Somewhere else on ruby core, the `:foo?` notation will

eventually come or has been discussed about, as there were other proposals and, I believe, no big objection against it (and probably the parser would not have a problem with it either) - nobu does not yet have 40 arms so he can not write all the code in a single day. :)

I think the difference between `:foo` and `:foo?` is quite minor. The difference between `:'foo-bar'` and `@'foo-bar'` seems a bit larger. Or even `@:'foo-bar'` ... the `->` and the `&`. lonely operator encourage me to use them for non-serious projects, together with the unicode snowman ... but I digress. :D

To the comparison given later:

```
instance_variable_set("@#{some_expression}", some_value)
"@#{some_expression}" = some_value
```

Well, version 2 is shorter, and often, being short and terse is nice, less code to write, hopefully also easy code that is easy to understand, and not too ugly.

But in this case, I like being explicit and it may confuse newcomers in the second variant; for instance, they may have to spot with their eyes that `@"#{some_expression}"` is different to `"#{@some_expression}"` or some other embedded construct. And I think when I see `&.->@"foo?"` it makes me wonder what language this is! :D

#3 - 03/17/2016 08:28 AM - shyouhei (Shyouhei Urabe)

At the developer's meeting yesterday we looked at this issue. As OP refers to `instance_variable_set`, this is a metaprogramming syntax by nature. And it seems there are no other syntax that does metaprogramming like this. Meeting attendees argued this point.

#4 - 03/17/2016 09:10 AM - sawa (Tsuyoshi Sawada)

I see. Thank you for bringing this issue to the meeting.

We can just have single quotation (without interpolation), and do not have the double quotation. Then, it would not be metaprogramming, I think. The main point is to extend the range of names allowed for instance/class variables.

But I also realize that, even if such things as `@'foo?'` were allowed, a method name like `foo?=` is still not allowed, so the advantage of my proposal may have been limited.

So I agree that this proposal may not have been a good idea anyway.

#5 - 05/09/2019 06:34 AM - sawa (Tsuyoshi Sawada)

I withdraw this feature request. Please close it.

#6 - 05/09/2019 06:49 AM - jeremyevans0 (Jeremy Evans)

- *Status changed from Open to Closed*