

## Ruby master - Feature #8765

### Literal for symbol with interpolation

08/10/2013 10:59 PM - sawa (Tsuyoshi Sawada)

<b>Status:</b>	Feedback
<b>Priority:</b>	Normal
<b>Assignee:</b>	
<b>Target version:</b>	
<b>Description</b>	
<pre>=begin I propose a symbol literal %S with interpolation  foo = 3 %S#{foo} # =&gt; :"\#{foo}" %S#{foo} # =&gt; : "3"  The conventional ":" literal looks as if it is a unary operator to a string. Since there are both %q and %Q literals for string, I don't see any reason why symbols only have %s.  =end</pre>	

### History

#### #1 - 08/11/2013 01:00 AM - duerst (Martin Dürst)

- Status changed from Open to Feedback

Do you need this in an actual program? There is %q and %Q because there is " and '", and that's because interpolation in Strings is very frequent. Interpolation for Symbols is very rare, as far as I know.

#### #2 - 08/11/2013 03:39 AM - sawa (Tsuyoshi Sawada)

```
=begin
There are ":" and ':' notations, whose existence indicate that symbol literals with and without interpolation are used. A typical use case for symbol literal is to create symbols from a list of words with a common prefix. For example, you may have a list of words
```

```
%w[foo bar baz]
```

and may want to create out of it a list of symbols with a common prefix:

```
%i[my_foo my_bar my_baz]
```

which, by the proposed syntax, can be created as:

```
%w[foo bar baz].map{|w| %S"my_#{w}"}
```

```
=end
```

#### #3 - 08/11/2013 01:59 PM - tenderlovmaking (Aaron Patterson)

On Sat, Aug 10, 2013 at 10:59:24PM +0900, sawa (Tsuyoshi Sawada) wrote:

Issue [#8765](#) has been reported by sawa (Tsuyoshi Sawada).

---

Feature [#8765](#): Literal for symbol with interpolation  
<https://bugs.ruby-lang.org/issues/8765>

Author: sawa (Tsuyoshi Sawada)

Status: Open

Priority: Normal

Assignee:

Category:

Target version:

```
=begin
I propose a symbol literal %S with interpolation
```

```
foo = 3
```

```
%s{#{foo}} # => :"\#{foo}"  
%S{#{foo}} # => :"3"
```

The conventional `:""` literal looks as if it is a unary operator to a string. Since there are both `%q` and `%Q` literals for string, I don't see any reason why symbols only have `%s`.

```
irb(main):001:0> RUBY_VERSION  
=> "2.0.0"  
irb(main):002:0> foo = 3  
=> 3  
irb(main):003:0> %i{#{foo}}  
=> [:"#{foo}"]  
irb(main):004:0> %l{#{foo}}  
=> [:"3"]  
irb(main):005:0>
```

```
--  
Aaron Patterson  
http://tenderlovmaking.com/
```

**#4 - 08/17/2013 10:54 PM - sawa (Tsuyoshi Sawada)**

tenderlovmaking

What is your intention?

**#5 - 08/19/2013 10:20 PM - Anonymous**

-1 here