

Ruby master - Feature #16994

Sets: shorthand for frozen sets of symbols / strings

06/26/2020 08:32 PM - marcandre (Marc-Andre Lafortune)

Status:	Feedback
Priority:	Normal
Assignee:	
Target version:	
Description	
I would like a shorthand syntax for <i>frozen Sets of symbols or of strings</i> .	
I am thinking of:	
<pre>%ws{hello world} # => Set['hello', 'world'].freeze %is{hello world} # => Set[:hello, :world].freeze</pre>	
The individual strings would be frozen. These literals would be created once at parse time (like Regex are):	
<pre>def foo p %ws{hello world}.object_id end foo foo # => prints the same id twice</pre>	
We should consider these sets to return a unique frozen to_a.	
Reminder: Ruby has literal notations for Rational and Complex. I've sadly never had to use either. I would venture to say that Complex is much less used than Sets, and that sets are underused.	
Reminder: previous discussion for builtin syntax was not for frozen literal, strings or symbols specifically: https://bugs.ruby-lang.org/issues/5478	
For builtin notations for generic sets (i.e. <i>unfrozen</i> or containing <i>other than string/symbol</i>), please discuss in another issue.	
Related issues:	
Related to Ruby master - Feature #16989: Sets: need ♥ <input type="checkbox"/>	Assigned

History

#1 - 06/26/2020 08:47 PM - marcandre (Marc-Andre Lafortune)

- Related to Feature #16989: Sets: need ♥ added

#2 - 08/24/2020 03:18 PM - Dan0042 (Daniel DeLorme)

+1

I think this is more important than having a general Set syntax as discussed in [#5478](#). Being able to use `%ws[foo bar].include?(str)` is a double-plus of not creating a new object each time and having O(1) efficiency.

#3 - 09/03/2020 03:03 AM - Dan0042 (Daniel DeLorme)

I just thought of something...

In the same way that `"str".freeze` is optimized to be deduplicated, `%w[a b].include?(obj)` could be optimized so it becomes equivalent to `obj == "a" || obj == "b"`, or something around those lines. This would have the advantage that all existing ruby code that uses this pattern would automatically become faster, without having to convert to a new literal syntax.

#4 - 09/03/2020 02:03 PM - Eregon (Benoit Daloze)

Dan0042 (Daniel DeLorme) wrote in [#note-3](#):

I just thought of something...

In the same way that `"str".freeze` is optimized to be deduplicated, `%w[a b].include?(obj)` could be optimized so it becomes equivalent to `obj == "a" || obj == "b"`, or something around those lines.

That already works on TruffleRuby (and for more than this specific case), it needs a JIT, inlining (also through builtins like `#include?`) and escape analysis.

#5 - 09/25/2020 04:26 AM - matz (Yukihiro Matsumoto)

- Status changed from Open to Feedback

We are going to introduce built-in set, but not in 3.0 (too little time to implement it before 3.0 release). After merging built-in set, we will seriously consider this proposal.

Remaining issues:

- Name? %ws would be the first two character specifier after %. Is it reasonable? Or should we seek another name?
- Frozen? %w returns non frozen array of non frozen strings. How should %ws behave?

Matz.

#6 - 09/25/2020 08:15 PM - normalperson (Eric Wong)

matz@ruby.or.jp wrote:

Remaining issues:

- Name? %ws would be the first two character specifier after %. Is it reasonable? Or should we seek another name?
- Frozen? %w returns non frozen array of non frozen strings. How should %ws behave?

How about suffix notation similar to Regexp modifiers?

```
[ 'foo', 'bar' ]s
```

Or with ability to specify ordering:

```
[ 'foo', 'bar' ]os # ordered set  
[ 'foo', 'bar' ]us # unordered set
```

Fwiw, I sometimes wish I could use unordered hash to save space:

```
{ 'foo' => 'bar' }u
```

And maybe 'f' modifier for frozen strings of values

<https://bugs.ruby-lang.org/issues/16994#change-87685>