Ruby master - Feature #18124
Hash shorthands (matching constructors functionality in JS)
08/20/2021 06:04 PM - myxoh (Nicolas Klein)

<table>
<thead>
<tr>
<th>Status:</th>
<th>Closed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority:</td>
<td>Normal</td>
</tr>
<tr>
<td>Assignee:</td>
<td></td>
</tr>
<tr>
<td>Target version:</td>
<td></td>
</tr>
</tbody>
</table>

### Description

**Suggestion:**

To implement one shorthand operators that allows you to construct hash into local variables and construct local variables into a hash.

### Context:

Javascript and other languages have a similar feature. It's helpful, and I don't think there's anything stopping Ruby for supporting it.  

Ruby also supports destructuring from an array.

### Syntax:

**Constructor:**

The constructor shorthand would allow you to create a hash where the symbol key has the same name as the variable.

For example:

```ruby
local_number = 1
user = OStruct.new(name: 'john')
hash = %C{ local_var user }
```

would be equivalent to:

```ruby
local_number = 1
user = OStruct.new(name: 'john')
hash = { local_number: local_number, user: user }
```

The new syntax is based on ruby's existing literal constructors syntax.

**EDIT:**

Edited to remove a reference to the destructuring, which is already implemented on Ruby 3

### Related issues:

- Related to Ruby master - Feature #14579: Hash value omission  Closed
- Related to Ruby master - Feature #17292: Hash Shorthand / Punning  Closed

### History

**#1 - 08/20/2021 06:18 PM - mame (Yusuke Endoh)**

- Related to Feature #14579: Hash value omission added

**#2 - 08/20/2021 06:18 PM - mame (Yusuke Endoh)**

- Related to Feature #17292: Hash Shorthand / Punning added

**#3 - 08/20/2021 06:26 PM - mame (Yusuke Endoh)**

For "Destructuring", you can use one-line pattern matching since Ruby 3.0:

```ruby
hash = {a: 'a', b: 'b'}
hash #=> { a: b: }
p hash[:a] #=> "a"
```
For "Constructor", it is a long-running topic: #14579, #17292, and maybe other tickets I cannot remember. But as far as I recall, no one proposed this particular syntax `%C{ }.

#4 - 08/20/2021 06:41 PM - myxoh (Nicolas Klein)
- Description updated
- Subject changed from Hash shorthands (matching constructors and destructuring functionality in JS) to Hash shorthands (matching constructors functionality in JS)

#5 - 08/20/2021 06:42 PM - myxoh (Nicolas Klein)
mame (Yusuke Endoh) wrote in #note-3:

For "Destructuring", you can use one-line pattern matching since Ruby 3.0:

```ruby
hash = {a: 'a', b: 'b'}
hash => { a:, b: }
p hash[:a] #=> "a"
```

For "Constructor", it is a long-running topic: #14579, #17292, and maybe other tickets I cannot remember. But as far as I recall, no one proposed this particular syntax `%C{ }.

Thank you, haven't had enough of a chance to play with Ruby 3. Clearly I missed things.
I've updated to remove the reference to the destructuring. I've also made a slight update to make it more consistent with ruby's existing % literal constructors (as that's the intent here)

#6 - 08/20/2021 06:43 PM - myxoh (Nicolas Klein)
- Description updated

#7 - 08/20/2021 06:46 PM - osyo (manga osyo)
mame (Yusuke Endoh) wrote in #note-3:

For "Destructuring", you can use one-line pattern matching since Ruby 3.0:

```ruby
hash = {a: 'a', b: 'b'}
hash => { a:, b: }
p hash[:a] #=> "a"
```

For "Constructor", it is a long-running topic: #14579, #17292, and maybe other tickets I cannot remember. But as far as I recall, no one proposed this particular syntax `%C{ }.

I think it's similar to this.
https://bugs.ruby-lang.org/issues/14973

#8 - 08/20/2021 06:52 PM - myxoh (Nicolas Klein)

osyo (manga osyo) wrote in #note-7:

mame (Yusuke Endoh) wrote in #note-3:

For "Destructuring", you can use one-line pattern matching since Ruby 3.0:

```ruby
hash = {a: 'a', b: 'b'}
hash => { a:, b: }
p hash[:a] #=> "a"
```

For "Constructor", it is a long-running topic: #14579, #17292, and maybe other tickets I cannot remember. But as far as I recall, no one proposed this particular syntax `%C{ }.

I think it's similar to this.
https://bugs.ruby-lang.org/issues/14973

Yes, fair find! If google translate is translating correctly it seems the exact same specification.
Sorry, should have done a better job searching for it!

Should I close this issue then?

#9 - 10/15/2021 06:39 PM - jeremyevans0 (Jeremy Evans)
- Status changed from Open to Closed