

Ruby - Feature #13137

Hash Shorthand

01/18/2017 09:11 PM - trevorlynnsmith (Trevor Smith)

Status:	Rejected	
Priority:	Normal	
Assignee:		
Target version:		
Description The ES6 update to Javascript added object shorthand. I propose that a similar hash shorthand be added to Ruby. Before <pre>name = 'Matz' country = 'Japan' attributes = { name: name, country: country }</pre> After <pre>name = 'Matz' country = 'Japan' attributes = { name, country }</pre> This would be very useful when building hashes from keyword arguments (especially with default values): <pre>def build(name: 'John Doe', age: 100, country: 'Earth',) attributes = { name, age, country } end</pre>		
Related issues: Related to Ruby - Feature #11105: ES6-like hash literals		Rejected

History

#1 - 01/20/2017 12:49 AM - znz (Kazuhiro NISHIYAMA)

- Related to Feature #11105: ES6-like hash literals added

#2 - 01/20/2017 01:27 AM - shevegen (Robert A. Heiler)

While I understand the primary reasoning behind it - less to type - I do not like the syntax proposal.

To me it looks as if there is something missing in the build-up of the hash elements.

It appears to put load onto my brain and appears to be inconsistent with how I would normally populate a hash with key-value pairs.

Perhaps one day we could use some advanced hash that does all of these things including hash with indifferent syntax.

But I think it would be better for core hash to stay the way it is without any new implicit syntax.

To quote matz though, he wrote in the other thread that there may be a chance to revisit it in the future, e. g. when ES6 syntax may be more popular.

#3 - 01/21/2017 06:36 PM - trevorlynnsmith (Trevor Smith)

My apologies – I searched extensively and did not find the previous feature request. Thank you for the consideration.

#4 - 03/13/2017 08:14 AM - matz (Yukihiro Matsumoto)

- *Status changed from Open to Rejected*

Rejected. I saw JavaScript new syntax, but I had no sympathy. It doesn't make anything more understandable.

Matz.

#5 - 04/13/2017 07:12 AM - naruse (Yui NARUSE)

Note: such culture seems to come from SGML.