

Ruby trunk - Feature #9980

Create HashWithIndifferentAccess using new syntax {a: 1}i

06/26/2014 12:23 PM - rosenfeld (Rodrigo Rosenfeld Rosas)

Status:	Rejected
Priority:	Normal
Assignee:	matz (Yukihiro Matsumoto)
Target version:	2.2.0
Description	
<p>This is related to feature #7797. For that feature I requested a regular hash ({} to be treated as HWIA and the current hash implementation would be available as StrictHash.new since it's not as much needed as the common case of expecting a[:a] to be the same as a['a'].</p> <p>Although I still prefer the proposal in #7797, I'd like to propose an alternative syntax for a new implementation of Hash that converts symbols to strings in keys (or keys to strings, it doesn't matter). This:</p> <pre>{my_key: 'my_value'}i</pre> <p>Should be equivalent to:</p> <pre>HashWithIndifferentAccess.new(my_key: 'my_value') # from ActiveSupport</pre> <p>It doesn't have to behave exactly like the implementation from ActiveSupport. I only really care about symbols and strings being treated the same way.</p> <p>This is not as great as #7797 because it will only affect user's own code rather than hashes created by other libraries (unless they decide to adopt the new syntax, which could take quite a while to keep backward compatibility with old Ruby versions). But it would already make it way more pleasant to define new hashes with indifferent access.</p> <p>Maybe it could act recursively during the hash creation, but I'm fine if you think it would be more confusing than helpful. To illustrate:</p> <pre>{a: {b: 1}}i</pre> <p>could mean the same as:</p> <pre>{a: {b: 1}i}i</pre>	

History

#1 - 06/26/2014 12:27 PM - rosenfeld (Rodrigo Rosenfeld Rosas)

- File *feature-9980.pdf* added

#2 - 06/30/2014 08:07 AM - naruse (Yui NARUSE)

received, thanks!

#3 - 06/30/2014 05:18 PM - rosenfeld (Rodrigo Rosenfeld Rosas)

- File *feature-9980.pdf* added

Reattaching using Firefox

#4 - 07/26/2014 04:58 AM - naruse (Yui NARUSE)

- File *deleted (feature-9980.pdf)*

#5 - 07/26/2014 05:12 AM - matz (Yukihiro Matsumoto)

- Status changed from Open to Rejected

Suffix `i` is used for complex (imaginary) numbers. It's bit confusing.
Maybe what you want can be gained by shorter/nicer name for `Hash#compare_by_identity`.

Matz.

#6 - 07/28/2014 12:56 PM - rosenfeld (Rodrigo Rosenfeld Rosas)

Was this rejected because the i suffix was already taken or because the idea of using suffixes would be confusing? Should I try another suffix in another ticket?

I don't think `compare_by_identity` does what I want. It seems to behave similar to ActiveSupport `Hash#symbolize_keys` which is different from `Hash#with_indifferent_access` which is what I really want.

I don't want to have to worry if I should call `hash[:a]` or `hash['a']`. Also `compare_by_identity` is a bad feature to me as it modifies the way the original hash works (it should be called `compare_by_identity!` in my opinion) and you can't undo its effect.

Maybe Ruby core could provide some utility to return a HWIA from a hash, like `H(a: 1)` meaning an equivalent of `HWIA.new(a: 1)`.

Files

feature-9980.pdf	24.8 KB	06/30/2014	rosenfeld (Rodrigo Rosenfeld Rosas)
------------------	---------	------------	-------------------------------------