Ruby trunk - Feature #8951

Please add a hash-to-hash alternative of the map method to Hash

09/25/2013 01:05 PM - behrangsa (Behrang Saeedzadeh)

Status:	Open
Priority:	Normal
Assignee:	
Target version:	

Description
Please see this thread: https://twitter.com/behrangsa/status/382670159086776323
Related issues:

Related to Ruby trunk - Feature #12512: Import Hash#transform_values and its ...	**Closed**

History

#1 - 09/25/2013 01:23 PM - fuadksd (Fuad Saud)

Wow, I missed this a couple of days ago. Totally supported.

--
Fuad Saud
Sent with Sparrow (http://www.sparrowmailapp.com/?sig)

On Wednesday, September 25, 2013 at 1:05 AM, behrangsa (Behrang Saeedzadeh) wrote:

> Issue #8951 has been reported by behrangsa (Behrang Saeedzadeh).
>
> _____
>
> Feature #8951: Please add a hash-to-hash alternative of the map method to Hash
> https://bugs.ruby-lang.org/issues/8951
>
> Author: behrangsa (Behrang Saeedzadeh)
> Status: Open
> Priority: Normal
> Assignee:
> Category:
> Target version:
>
> Please see this thread: https://twitter.com/behrangsa/status/382670159086776323
>
> --
> http://bugs.ruby-lang.org/

#2 - 09/25/2013 01:36 PM - jamonholmgren (Jamon Holmgren)

How about:

Hash#graph

or

Hash#rehash

#3 - 09/25/2013 02:23 PM - fuadksd (Fuad Saud)

#rehash is taken: http://www.ruby-doc.org/core-2.0.0/Hash.html#method-i-rehash

I was thinking about 'projection' or something, but everything feels just like an alternative to 'map' (#map, in my opinion, should return a hash).

--
Fuad Saud
Sent with Sparrow (http://www.sparrowmailapp.com/?sig)

On Wednesday, September 25, 2013 at 1:36 AM, jamonholmgren (Jamon Holmgren) wrote:

> Issue #8951 has been updated by jamonholmgren (Jamon Holmgren).

How about:

Hash#graph

or

Hash#rehash

Feature #8951: Please add a hash-to-hash alternative of the map method to Hash
https://bugs.ruby-lang.org/issues/8951#change-41971

Author: behrangsa (Behrang Saeedzadeh)
Status: Open
Priority: Normal
Assignee:
Category:
Target version:

Please see this thread: https://twitter.com/behrangsa/status/382670159086776323

--
http://bugs.ruby-lang.org/

#4 - 09/26/2013 09:07 AM - Anonymous

In my private library, I have 3 map-like methods defined on hash:

Hash#with_keys
Hash#with_values
Hash#modify

They work like this:

h = { a: 1, b: 2 }
h.with_keys do |k| k.to_s end #=> { "a"=>1, "b"=>2 }
h.with_values do |v| v.to_s end #=> { a: "1", b: "2" }
h.modify do |k, v| [k, "#{k}#{v}"] end #=> { a: "a1", b: "b2" }

plus their ! versions that modify the hash in place:

Hash#with_keys!, Hash#with_values!, Hash#modify!

But, this much being said, I am not sure whether this feature creep is such a good thing. Part of the fun in Ruby is that you are inventing your own libraries. If everything simple is already done in advance, it will be more "work" memorizing and less "fun" inventing :-))) Just randomly saying whatever comes to my mind, don't take me too seriously :-)

#5 - 10/13/2013 10:11 PM - behrangsa (Behrang Saeedzadeh)

How about having a construct similar to map.with_index? For example map.to_has or map.hashify:

```
hash = { ... }
hash.map.to_hash do |e|
 convert(e)
end
```

or

```
hash = { ... }
hash.map.hashify do |e|
 convert(e)
end
```

#6 - 12/03/2013 11:10 AM - fuadksd (Fuad Saud)

Hash#transform might also be a good name.

#7 - 06/16/2014 03:28 PM - Ajedi32 (Andrew M)

I believe this is a duplicate of #6669

#8 - 06/22/2016 12:14 AM - shyouhei (Shyouhei Urabe)

- Related to Feature #12512: Import Hash#transform_values and its destructive version from ActiveSupport added