

Ruby trunk - Feature #4772

Hash#add_keys

05/24/2011 05:36 PM - yimutang (Joey Zhou)

Status:	Rejected
Priority:	Normal
Assignee:	matz (Yukihiro Matsumoto)
Target version:	2.6
Description	
Hi, do you want to add a new method Hash#add_keys in a new version?	
<pre>hash = Hash.new { h,k k.to_s + "foo" } hash.add_keys("a","b","c") # the value is hash's default obj or proc value</pre>	
If there's a word list file, I want to make the words keys of a hash, maybe I can write:	
<pre>hash = {} hash.add_keys(*open("file").readlines.map(&:chomp))</pre>	

History

#1 - 05/25/2011 01:30 AM - matz (Yukihiro Matsumoto)

I see no reason to add a method to generate keys only. Hash is a set of key-value pairs, right? Can you elaborate why you want this method?

matz.

#2 - 05/25/2011 06:29 PM - aprescott (Adam Prescott)

On Tue, May 24, 2011 at 9:36 AM, Joey Zhou yimutang@gmail.com wrote:

```
hash = Hash.new {|h,k| k.to_s + "foo" }
hash.add_keys("a","b","c") # the value is hash's default obj or proc value
```

If there's a default proc, I can't see why you'd need to create the keys like this. If you did want to, can you not just use

```
keys.each { |k| hash[keys] }
```

?

For instance:

```
open("file").readlines.map(&:chomp).each { |k| hash[k] }
```

#3 - 05/25/2011 10:35 PM - yimutang (Joey Zhou)

Yukihiro Matsumoto wrote:

I see no reason to add a method to generate keys only. Hash is a set of key-value pairs, right? Can you elaborate why you want this method?

matz.

Maybe the method name leads to misunderstanding. What I mean is that I give keys, the hash generate values automatically, not generating keys only.

For example:

```
hash = Hash.new {|h,k| k + k.succ}
hash.add_keys("a","b","c")
```

Now, the hash should be {"a"=>"ab","b"=>"bc","c"=>"cd"}

In recent version, if I want to do the same thing, the code may be:

```
hash = Hash.new {|h,k| k + k.succ}
```

```
["a","b","c"].each do |key|
hash[key] = hash[key] # this kind of assignment is somewhat odd
end
```

```
hash = Hash.new {|h,k| h[k] = k + k.succ}
["a","b","c"].each do |key|
hash[key] # making key appears means making it exists? odd too.
end
```

So I think a method "filling hash with given keys and auto-generated values" is OK.

The name and parameter detail may be thought over.

Joey

#4 - 05/25/2011 10:38 PM - yimutang (Joey Zhou)

Adam Prescott wrote:

On Tue, May 24, 2011 at 9:36 AM, Joey Zhou yimutang@gmail.com wrote:

```
hash = Hash.new {|h,k| k.to_s + "foo" }
hash.add_keys("a","b","c") # the value is hash's default obj or proc value
```

If there's a default proc, I can't see why you'd need to create the keys like this. If you did want to, can you not just use

```
keys.each { |k| hash[keys] }
```

?

For instance:

```
open("file").readlines.map(&:chomp).each { |k| hash[k] }
```

This doesn't create the actual key/value pair, the way seems not explicit, I'm afraid.

#5 - 05/25/2011 10:44 PM - rosenfeld (Rodrigo Rosenfeld Rosas)

In recent version, if I want to do the same thing, the code may be:

```
hash = Hash.new {|h,k| k + k.succ}
["a","b","c"].each do |key|
hash[key] = hash[key] # this kind of assignment is somewhat odd
end
```

Actually, as already noticed here, this could be just:

```
['a', 'b', 'c'].each {|k| hash[k]} # You don't need to assign to it.
```

But I like your suggestion anyway...

#6 - 05/25/2011 10:59 PM - mame (Yusuke Endoh)

Hello,

2011/5/25 Rodrigo Rosenfeld Rosas rr.rosas@gmail.com:

Actually, as already noticed here, this could be just:

```
['a', 'b', 'c'].each {|k| hash[k]} # You don't need to assign to it.
```

The assignment is needed.

```
hash = Hash.new {|h,k| k + k.succ}
["a","b","c"].each {|k| hash[k] }
```

```
p hash #=> {} <- LOOK!
```

This feature might be useful to use a hash as a set...
I'm not sure though.

--

Yusuke Endoh mame@tsg.ne.jp

#7 - 05/25/2011 11:20 PM - yimutang (Joey Zhou)

Rodrigo Rosenfeld Rosas wrote:

Actually, as already noticed here, this could be just:

```
['a', 'b', 'c'].each {|k| hash[k]} # You don't need to assign to it.
```

Well, it is very obscure here, easy to be confused.

```
1) hash = Hash.new {|h,k| k.succ }  
2) hash = Hash.new {|h,k| h[k] = k.succ }
```

1) and 2) are different, 2) will actually create any key/value pair the hash ever seen a key, but 1) will not, it needs an assignment.

Joey

#8 - 05/25/2011 11:23 PM - rosenfeld (Rodrigo Rosenfeld Rosas)

Em 25-05-2011 10:54, Yusuke ENDOH escreveu:

Hello,

2011/5/25 Rodrigo Rosenfeld Rosas rr.rosas@gmail.com:

Actually, as already noticed here, this could be just:

```
['a', 'b', 'c'].each {|k| hash[k]} # You don't need to assign to it.
```

The assignment is needed.

```
hash = Hash.new {|h,k| k + k.succ}  
["a","b","c"].each {|k| hash[k]}
```

```
p hash #=> {}<- LOOK!
```

This feature might be useful to use a hash as a set...
I'm not sure though.

Yes, I didn't read the documentation and believed on Adam statement, sorry :)

I think that "add_keys" makes the code more readable than the current alternative.

Or maybe "add_keys_with_default_value" :)

#9 - 05/26/2011 04:23 AM - aprescott (Adam Prescott)

On Wed, May 25, 2011 at 3:20 PM, Joey Zhou yimutang@gmail.com wrote:

```
1) hash = Hash.new {|h,k| k.succ }  
2) hash = Hash.new {|h,k| h[k] = k.succ }
```

1) and 2) are different, 2) will actually create any key/value pair the hash ever seen a key, but 1) will not, it needs an assignment.

Ah I see, I missed the lack of assignment in the block to Hash.new. I suppose potentially there's some confusion that might come up when using Hash#add_keys and Hash.new given that one block is being used for both purposes?

#10 - 03/25/2012 03:55 PM - mame (Yusuke Endoh)

- Status changed from Open to Assigned

- Assignee set to matz (Yukihiro Matsumoto)

#11 - 11/20/2012 09:38 PM - mame (Yusuke Endoh)

- Target version set to 2.6

#12 - 11/20/2012 10:49 PM - matz (Yukihiro Matsumoto)

- Status changed from Assigned to Rejected

As far as I understand, the OP intention was to generate key/default-value pair explicitly.
But I don't see any real world use case. The value will be generated anyway.
If he wants to cache the value, the name #add_keys does not describe the intention.

So I mark this "rejected". If OP (or somebody else) come up with real use case, please re-open.

Matz.