

Ruby master - Misc #17053

RDoc for Hash Keys

07/27/2020 03:42 PM - burdettelamar@yahoo.com (Burdette Lamar)

Status:	Open
Priority:	Normal
Assignee:	
Description	
marcandre comments on my pull request regarding documentation of Hash in Rdoc :	
<p>The only thing I would change is that I would shorten the doc on the "Invalid Hash Keys". As far as I know, this is simply not a[n] important concern as nearly all Ruby objects respond_to? :hash and :eql?</p> <p>I personally would recommend adding a single example in the Hash.html#class-Hash-label-Hash+Keys section and I would remove the rest, or at least remove the examples. They burden the reader with something that is of no use to them.</p> <p>I have misgivings about it:</p> <ul style="list-style-type: none">• Some of the material, for example, the text and example for user-defined hash keys, is very old.• I consolidated some material from earlier doc for individual methods, which now link to the relevant sections.• All are factual and not repeated elsewhere in the page. <p>This is an API reference documentation. Ruby should have a <i>reference documentation</i>, and therefore should not omit anything.</p> <p>If such material is to be included, I see three possibilities:</p> <ul style="list-style-type: none">• Include inline, as is now.• Link to a footnote on the same page, with a back link.• Link to another rdoc page in doc/ dir. <p>I'd love to hear some opinions on this.</p>	

History

#1 - 07/27/2020 03:43 PM - burdettelamar@yahoo.com (Burdette Lamar)

- Backport deleted (2.5: UNKNOWN, 2.6: UNKNOWN, 2.7: UNKNOWN)

- ruby -v deleted (ruby 2.7.0p0 (2019-12-25 revision 647ee6f091) [x64-mingw32])

- Tracker changed from Bug to Misc

#2 - 07/27/2020 04:36 PM - sawa (Tsuyoshi Sawada)

Where is the source of the quote? Neither of the links seems to lead to it.

#3 - 07/27/2020 05:55 PM - burdettelamar@yahoo.com (Burdette Lamar)

Should have linked to the source of the remarks: <https://github.com/ruby/ruby/pull/3139#issuecomment-663281047>

#4 - 07/27/2020 06:37 PM - marcandre (Marc-Andre Lafortune)

Let me first repeat that I am very grateful of all efforts to improve our documentation.

Documentation can be difficult because it requires deep understanding of what needs to be documented and good experience with what is actually useful for users.

What I am referring to is that the documentation for, e.g Hash#values_at, currently reads as:

```
call-seq:
  hash.values_at(*keys) -> new_array
Returns a new \Array containing values for the given +keys+:
  h = {foo: 0, bar: 1, baz: 2}
  h.values_at(:foo, :baz) # => [0, 2]
Returns an empty Array if no arguments given:
  h = {foo: 0, bar: 1, baz: 2}
  h.values_at # => []
```

```
Raises an exception if any given key is invalid
(see {Invalid Hash Keys}[#class-Hash-label-Invalid+Hash+Keys]):
  h = {foo: 0, bar: 1, baz: 2}
  # Raises NoMethodError (undefined method `hash' for #<BasicObject>):
```

There are 3 examples.

The first one is generic.

The second one is a corner case that is to me feels both obvious and not useful, but maybe it isn't for everybody. I didn't even try suggesting to remove it, although I see little value in it.

What I've asked @burnettelamar to do (or offered to do myself) was to remove the last example and simply refer to the top-level documentation for invalid keys, because that example feels to me as a burden on the reader that is not helpful.

Moreover, the sentence "Raises an exception if any given key is invalid" is not even accurate:

```
# empty hash case
{}.values_at(BasicObject.new) # => [nil]
# or
h = {a: 1, b: 2}
h.compare_by_identity
h.values_at(BasicObject.new) # => [nil]
```

The invalid key section seems to imply that keys need to implement hash, which also is inaccurate as implementing eql? is also mandatory.

Even if it was accurate, my feeling is that trying to use invalid keys in a hash is not a concern and should not be addressed repeatedly with examples for values_at, slice, etc.

My experience is that very very few Rubyists use BasicObject or objects that do not define eql? and hash.

Finally, the current documentation gives no example of values_at with a key that isn't found, and doesn't even state that the default/default_proc is used in those cases, which I believe to be useful.

I am surprised that the above, once pointed out, isn't clear.

#5 - 07/27/2020 07:22 PM - marcandre (Marc-Andre Lafortune)

burdettelamar@yahoo.com (Burdette Lamar) wrote:

My view has been this: This is API reference documentation. Ruby/ruby should have *the reference documentation*, and therefore should omit nothing.

This is a very ambitious goal I'm not sure I share completely. Taking for example the documentation for Hash, one would need to talk about covariance of methods returning a hash (i.e. Class.new(Hash).new.select{}.class vs Class.new(Hash).new.merge({}).class), of treatment of a key Float::NAN (which is not eql? to itself), of recursive hashes, of the arity of enumerators (see <https://bugs.ruby-lang.org/issues/14015#note-8>), of the performance of hash lookup / insertion...

All these details can not be repeated for each method, the same way we can't quote in full the floating point standard for Float#+ and repeat it for Float#-, etc. On that subject, an example with 0.1 + 0.2 might be helpful though.

#6 - 07/27/2020 07:36 PM - burdettelamar@yahoo.com (Burdette Lamar)

Thanks, @marcandre. Your first update above makes things much clearer to me. I'll continue studying this.

#7 - 07/28/2020 01:21 AM - sawa (Tsuyoshi Sawada)

- Description updated