Ruby - Feature #10130

String format with nested hash

08/14/2014 12:10 AM - sawa (Tsuyoshi Sawada)

Status:	Rejected	
Priority:	Normal	
Assignee:	matz (Yukihiro Matsumoto)	
Target version:		

Description

When we do string format with hashes, we have the restriction that the hash cannot be a nested one. When we want to handle more complicated string formats, for example in templates, it is more convenient if we can use a nested hash.

```
"Author: %{author.name} (%{author.affiliation}), %{date}" % {author: {name: "Ruby Taro", affiliati
on: "Ruby co."}, date: "2014, 8, 14"}
#=> "Author: Ruby Taro (Ruby co.), 2014, 8, 14"
```

History

#1 - 08/14/2014 12:26 AM - avit (Andrew Vit)

I would have expected [] for hash syntax in the string template:

```
"Author: %{author[:name]} (%{author[:affiliation]}), %{date}"
```

For it to work with dots seems inconsistent, unless the hash value responds to those methods.

#2 - 08/14/2014 02:12 AM - matz (Yukihiro Matsumoto)

- Category set to core

- Status changed from Open to Rejected

- Assignee set to matz (Yukihiro Matsumoto)

Rejected, for several reasons:

(1) author.name is not a canonical way to access nested hash.
(2) this feature add more complexity than it gains.
(3) you can just use #{author[:name]} to embed.

Matz.

#3 - 12/12/2015 01:29 PM - sawa (Tsuyoshi Sawada)

Now that we have Hash#dig and Array#dig coming for Ruby 2.3, I think that this proposal of mine from the past makes more sense.

Regarding Matz' point (2), this proposal should parallel the feature of dig. It doesn't add extra cognitive load to us beyond what we would be knowing about dig in the coming version of Ruby.

Regarding Matz' point (3), string format (%{}) cannot be replaced by string interpolation (#{}) when we want a static string that is to be used as a template, to which values would be inserted later to output a page. Interpolation requires dynamic evaluation of the string each time.

Regarding Matz' point (1), instead of using the period as the delimiter, we can use a comma (followed by optional space characters), which would resemble how dig is called.

```
"Author: %{author, name} (%{author, affiliation}), %{date}" % {author: {name: "Ruby Taro", affiliation: "Ruby
co."}, date: "2014, 8, 14"}
#=> "Author: Ruby Taro (Ruby co.), 2014, 8, 14"
```

This requires a key with a comma inside %{} not to be interpreted as a single symbol but as a sequence of symbols separated by the comma (followed by space characters). Since it is rare that a comma is used in a symbol, I don't think it will affect existing code.

#4 - 12/20/2015 12:01 PM - sawa (Tsuyoshi Sawada)

What about using a sequence of space characters as delimiter?

```
h = {
  author: {
    name: "Ruby Taro",
```

```
    affiliation: "Ruby co.",
  },
  date: "2014, 8, 14"
}

"Author: %{author name} (%{author affiliation}), %{date}" % h
#=> "Author: Ruby Taro (Ruby co.), 2014, 8, 14"

"%{author name} works at %{author affiliation}" % h
#=> "Ruby Taro works at Ruby co."
```

This is reminiscent of (and hence consistent with) %w{} and %i{} notations, which also use a sequence of space characters as delimiter.