

Ruby master - Feature #14417

String#sub / String#gsub `Symbol` `Hash`

01/29/2018 09:35 AM - osyo (manga osyo)

Status:	Feedback
Priority:	Normal
Assignee:	
Target version:	

Description

String#sub / String#gsub `Symbol` `Hash` `String`

```
hash = {'b'=>'B', 'c'=>'C'}
p "abcabc".gsub(/[bc]/, hash) #=> "aBCaBC"
```

```
# Symbol Hash
hash = { b: 'B', c: 'C' }
p "abcabc".gsub(/[bc]/, hash) #=> "aa"
```

```
# Symbol Hash String
p "abcabc".gsub(/[bc]/, hash.transform_keys(&:to_s)) #=> "aBCaBC"
```

```
# String
hash = {'b'=>'B', 'c'=>'C'}
p "abcabc".gsub(/[bc]/, hash) #=> "aBCaBC"
```

```
hash = { b: 'B', c: 'C' }
```

```
# $& String
p "abcabc".gsub(/[bc]/){hash[$&]} #=> "aa"
```

```
# String
p "abcabc".gsub(/[bc]/){ |s| hash[s] } #=> "a"
```

```
# Hash Symbol
p "abcabc".gsub(/[bc]/, hash) #=> "aBCaBC"
```

- `Symbol` `Hash`
 - [ruby-style-guide/README.ja.md at japanese · fortissimo1997/ruby-style-guide · GitHub](#)
- `Symbol` `Hash`

- `String` `Symbol`
 - "abcabc".sub(/[bc]/, { "b" => "A", b: "C" }) # => ???
 - `String`
 - `String` `Symbol` `Hash`
 - `String`

String#gsub `Symbol` `Hash`

```
# http://batsov.com/articles/2013/10/03/using-rubys-gsub-with-a-hash/
def geekify(string)
```


Thanks for reply!!!

would it probably better if sub/gsub would call hash.transform_keys(&.to_s) internal in their code with the hash if hash is given?

hmmmm... Thanks for idea :)

XX

XXXXXXXX...XX
XXXXXXXXXXXXXXXXXXXXXXXXXXXX

#4 - 01/29/2018 03:17 PM - Hanmac (Hans Mackowiak)

did look at string.c for gsub code,
<https://github.com/ruby/ruby/blob/trunk/string.c#L5094>
seems to be the line where we could add a transform_keys call

but i don't know currently what the best way to call hash.transform_keys(&.to_s)
probably something with rb_funcall_with_block?

#5 - 01/31/2018 11:06 AM - duerst (Martin Dürst)

gsub with Hash is used in some contexts where high performance is of interest. An example is lib/unicode_normalize/normalize.rb. This proposal would make these cases less efficient, for the benefit of people who can't keep Symbols and Strings apart.

As discussed in another issue, b: 'B', c: 'C' is not a shortcut for 'b'=>'B', 'c'=>'C'. We already have methods to change Hash keys (or values), and we probably need more of them, but I think we don't need more methods that accepts strings and symbols indeterminately.

#6 - 01/31/2018 11:17 AM - Hanmac (Hans Mackowiak)

[duerst \(Martin Dürst\)](#): what about my example where it does transform the key internal for the given Hash?
or is that a nono too?
it might be possible to only do it if the given hash has non String key?

#7 - 02/04/2018 01:05 PM - naruse (Yui NARUSE)

Hanmac (Hans Mackowiak) wrote:

[duerst \(Martin Dürst\)](#): what about my example where it does transform the key internal for the given Hash?
or is that a nono too?
it might be possible to only do it if the given hash has non String key?

If the hash is called many times from gsub, those integers shold be converted as String before gsub.
Because such conversion needs object allocation many times, and cause many GC.

I think

```
h = {1 => "A", 2 => "B", 3 => "C", 4 => "D", 5 => "E"}  
"12345".gsub(/\d/){ h[$&.to_i] }
```

is faster than such code.

Files

string_sub_with_symbol_key.patch	2.71 KB	01/29/2018	osyo (manga osyo)
----------------------------------	---------	------------	-------------------