

## Ruby master - Bug #8512

### gsub() works incorrect

06/11/2013 06:30 PM - gainaktar (Oleg K)

<b>Status:</b> Rejected	
<b>Priority:</b> Normal	
<b>Assignee:</b>	
<b>Target version:</b>	
<b>ruby -v:</b> ruby 2.0.0p195 (2013-05-14) [i386-mingw32]	<b>Backport:</b> 1.9.3: UNKNOWN, 2.0.0: UNKNOWN
<b>Description</b> irb(main):005:0> "\.gsub("\\", "\\").length => 1 irb(main):006:0> "\.gsub("\\", "XX").length => 2  bug is duplicated with rejected bug <a href="#">#8511</a>	

### History

#### #1 - 06/11/2013 09:41 PM - Eregon (Benoit Daloze)

- Status changed from Open to Rejected

This is due to Regexp replace syntax and literal strings.

In literal strings, you need two \ to produce one \ character (a single is the start of an escape character like \t, \n, ...).

And in Regexp replacement strings, you need to escape the \ (a single one is the beginning of a special replacement sequence like \1,&,...).

So that makes 4 \ for one produced in literal replacement strings:

```
\".gsub("\\", "\\").length  
=> 1  
\".gsub("\\", "\\").length  
=> \"
```

Not so nice, but definitely expected behavior.

#### #2 - 06/11/2013 09:45 PM - Eregon (Benoit Daloze)

This should be documented in Regexp's overview though.

#### #3 - 06/11/2013 09:45 PM - Hanmac (Hans Mackowiak)

=begin

the docs says:

*((If replacement is a String it will be substituted for the matched text. It may contain back-references to the pattern's capture groups of the form \\d, where d is a group number, or \\k, where n is a group name. If it is a double-quoted string, both back-references must be preceded by an additional backslash.))*

so you need more \" in your string

=end

#### #4 - 06/11/2013 10:15 PM - gainaktar (Oleg K)

Hanmac (Hans Mackowiak) wrote:

=begin

the docs says:

*((If replacement is a String it will be substituted for the matched text. It may contain back-references to the pattern's capture*

*groups of the form \\d, where d is a group number, or \\k, where n is a group name. If it is a double-quoted string, both back-references must be preceded by an additional backslash.))*

so you need more "\" in your string  
=end

Thanks for the explanation.

**#5 - 06/11/2013 10:35 PM - phluid61 (Matthew Kerwin)**

For the record, you can also use the hash or block replacement forms, because those doesn't use regexp back-references:

```
"\".gsub("\\", "\\>\\") #=> "\"  
"\".gsub("\\") { "\\\" } #=> "\"
```