

## Ruby - Bug #7042

### String#=~ is not equal to Regexp#=~

09/21/2012 02:00 PM - Hanmac (Hans Mackowiak)

<b>Status:</b>	Rejected	<b>Backport:</b>
<b>Priority:</b>	Normal	
<b>Assignee:</b>		
<b>Target version:</b>		
<b>ruby -v:</b>	ruby 1.9.3p194 (2012-04-20 revision 35410) [x86_64-linux]	
<b>Description</b>		
<p>i found out that string =~ regexp is not equal to regexp =~ string when you use named captures:</p> <p>"6:15 PM" =~ /(?\d+):(?\d+) (?A P)M/; p hour,minute,pm # works not /(?\d+):(?\d+) (?A P)M/ =~ "6:15 AM"; p hour,minute,pm # works</p> <p>i wonder why is that so ?</p>		

### History

#### #1 - 09/23/2012 10:40 PM - stomar (Marcus Stollsteimer)

=begin  
From the documentation for Regexp:

"When named capture groups are used with a literal regexp on the left-hand side of an expression and the =~ operator, the captured text is also assigned to local variables with corresponding names."

Of course, this does not really answer the question regarding the "why".

Regards,  
Marcus  
=end

#### #2 - 09/24/2012 04:24 AM - stomar (Marcus Stollsteimer)

=begin  
...and further down in the Regexp docs:

"This assignment is implemented in the Ruby parser. The parser detects 'regexp-literal =~ expression' for the assignment. The regexp must be a literal without interpolation and placed at left hand side."

"The assignment does not occur if the regexp is not a literal."  
=end

#### #3 - 11/05/2012 08:38 PM - mame (Yusuke Endoh)

- Status changed from Open to Rejected

Because this is a kind of assignment, the value is in the right side and the variable is in the left side... we thought.

This is actually intentional. Closing.  
You can open a feature request to add a pattern to regexp assignment, if you wish.

--  
Yusuke Endoh [mame@tsg.ne.jp](mailto:mame@tsg.ne.jp)