

Ruby master - Bug #12862

Regular Expression Named Group Matching does not work with #{} (Similar? to issue #2778)

10/22/2016 03:46 AM - Amig0 (Leo Amigud)

Status: Closed	
Priority: Normal	
Assignee:	
Target version:	
ruby -v: ruby 2.3.1p112 (2016-04-26 revision 54768) [x86_64-linux]	Backport: 2.1: UNKNOWN, 2.2: UNKNOWN, 2.3: UNKNOWN

Description

Loading development environment (Rails 4.2.5)

```
irb(main):001:0> /\$(?<dollars>\d+)\.(?<cents>\d+)/ =~ "$3.67"  
=> 0  
irb(main):002:0> dollars  
=> "3"
```

BUT:

```
irb(main):001:0> numbers = '\d+'  
=> "\\d+"  
irb(main):002:0> /\$(?<dollars>#{numbers})\.(?<cents>\d+)/ =~ "$3.67"  
=> 0  
irb(main):003:0> dollars  
NameError: undefined local variable or method `dollars' for main:Object
```

History

#1 - 10/22/2016 03:51 PM - Hanmac (Hans Mackowiak)

it did took me a while to understand what you mean, because the regexp are displayed wrong.

"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."

it might not be good explained, but that literal regexp can not have interpolation for that to work.

#2 - 10/22/2016 10:59 PM - Amig0 (Leo Amigud)

Hans Mackowiak wrote:

it did took me a while to understand what you mean, because the regexp are displayed wrong.

"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."

it might not be good explained, but that literal regexp can not have interpolation for that to work.

Sorry, not sure what you mean.

As a matter of fact i can use regexp like:

```
irb(main):013:0> numbers = '\d+'  
=> "\\d+"  
irb(main):014:0> /\$#{numbers}\d+/ =~ "$3.67"  
=> 0  
irb(main):015:0>
```

to match the string, but using #{numbers} inside the named group somehow does not work.

Is that the expected behaviour?

Leo

#3 - 10/22/2016 11:21 PM - phluid61 (Matthew Kerwin)

Leo Amigud wrote:

Sorry, not sure what you mean.

As a matter of fact i can use regexp like:

```
irb(main):013:0> numbers = 'd+'
=> "d+"
irb(main):014:0> /\#{numbers}.d+/ =~ "$3.67"
=> 0
irb(main):015:0>
```

to match the string, but using `\#{numbers}` inside the named group somehow does not work.

Is that the expected behaviour?

Leo

It's what Hans quoted and said:

"When named capture groups are used **with a literal regexp** [*i.e. not an interpolated regexp*] ... the captured text is also assigned to local variables ..."

`/(?<dollars>\#{numbers})/` is interpolated, not literal.

If you want to use interpolation and named captures, you can still get there using `#match`:

```
irb(main):001:0> numbers = 'd+'
=> "\\d+"
irb(main):002:0> md = /\$(?<dollars>(\#{numbers})).(?<cents>\d+)/.match "$4.67"
=> #<MatchData "$4.67" dollars:"4" cents:"67">
irb(main):003:0> md['dollars']
=> "4"
irb(main):004:0> md['cents']
=> "67"
```

or:

```
irb(main):001:0> numbers = 'd+'
=> "\\d+"
irb(main):002:0> /\$(?<dollars>(\#{numbers})).(?<cents>\d+)/ =~ "$4.67"
=> 0
irb(main):003:0> $~['dollars']
=> "4"
irb(main):004:0> $~['cents']
=> "67"
```

#4 - 10/23/2016 02:48 AM - Amig0 (Leo Amigud)

Matthew Kerwin wrote:

Leo Amigud wrote:

Sorry, not sure what you mean.

As a matter of fact i can use regexp like:

```
irb(main):013:0> numbers = 'd+'
=> "d+"
irb(main):014:0> /\#{numbers}.d+/ =~ "$3.67"
=> 0
irb(main):015:0>
```

to match the string, but using `\#{numbers}` inside the named group somehow does not work.

Is that the expected behaviour?

Leo

It's what Hans quoted and said:

"When named capture groups are used **with a literal regexp** [*i.e. not an interpolated regexp*] ... the captured text is also assigned to local

variables ..."

`/(?<dollars>#{numbers})/` is interpolated, not literal.

If you want to use interpolation and named captures, you can still get there using `#match`:

```
irb(main):001:0> numbers = '\d+'
=> "\\d+"
irb(main):002:0> md = /\$(?<dollars>#{numbers})\.(?<cents>\d+)/.match "$4.67"
=> #<MatchData "$4.67" dollars:"4" cents:"67">
irb(main):003:0> md['dollars']
=> "4"
irb(main):004:0> md['cents']
=> "67"
```

or:

```
irb(main):001:0> numbers = '\d+'
=> "\\d+"
irb(main):002:0> /\$(?<dollars>#{numbers})\.(?<cents>\d+)/ =~ "$4.67"
=> 0
irb(main):003:0> $~['dollars']
=> "4"
irb(main):004:0> $~['cents']
=> "67"
```

Thank you for the explanation!

Leo

#5 - 10/23/2016 11:22 AM - shyouhei (Shyouhei Urabe)

- Description updated

#6 - 10/18/2019 12:48 AM - jeremyevans0 (Jeremy Evans)

- Status changed from Open to Closed