

Programming Puzzles & Code Golf Stack Exchange works best with JavaScript enabled

8 +10

join this community tour help

Programming Puzzles & Code Golf Stack Exchange is a question and answer site for programming puzzle enthusiasts and code golfers. Join them; it only takes a minute:

Here's how it works:

Anybody can ask a question

Anybody can answer

The best answers are voted up and rise to the top

Regex that only matches itself

There are some pretty cool challenges out there involving regex ([Self-matching regex](#), [Regex validating regex](#))

This may well be impossible, but is there a regex that will ONLY match itself?

NOTE, delimiters must be included:

for example `/thing/` must match `/thing/` and not `thing`. The only match possible for your expression must be the expression itself. Many languages allow the implementation of a string in the place of a regular expression. [For instance in Go](#)

```
package main

import "fmt"
import "regexp"

func main() {

    var foo = regexp.MustCompile("bar")
    fmt.Println(foo.MatchString("foobar"))
}
```

but for the sake of the challenge, let the expression be delimited (starting symbol, expression, ending symbol ex: `/fancypantpattern/` or `@[^2048]@`), if you want to argue quotes as your delimiter, so be it. I think given the apparent difficulty of this problem it won't make much of a difference.

To help you along:

Quick hack I put together for [rubular.com](#) (a webpage for ruby regex editing):

```
var test = document.getElementById("test")
,regex = document.getElementById("regex")
,delimiter="/"
,options = document.getElementById("options")
,delay = function(){test.value = delimiter + regex.value + delimiter + options.value}
,update = function(e){
    // without delay value = not updated value
    window.setTimeout(delay,0);
}
regex.onkeydown = update;
options.onkeydown = update;
```

Even though this is technically 'code golf' I will be very impressed if anyone can find an answer/ prove it is impossible.

Link is now fixed. Sorry to all

Winning answer thus far: [jimmy23013](#) with 40 characters

[code-golf](#) [string](#) [regular-expression](#)

edited yesterday

Purplejacket
103 2

asked May 30 '14 at 16:20

Dylan Madisetti
1,368 2 6 13

1 Obviously any regular expression that only includes literals will work: `//`, `/a/`, `/xyz/`, etc. It might be good to require that the regex has to include a non-literal operation. – [breadbox](#) May 30 '14 at 16:22

6 literals won't work because you're required to match the backslashes for example `/aaa/` will match `aaa` but not `/aaa/` – [Dylan Madisetti](#) May 30 '14 at 16:28

1 @DylanMadisetti Do we have to use `//` delimiters, or can we choose other delimiters (PCRE supports pretty much character, and in particular you can use matched parentheses/braces/brackets as delimiters). – [Martin Ender](#) ♦ May 30 '14 at 19:18

Programming Puzzles & Code Golf Stack Exchange works best with JavaScript enabled

If $\{m,n\}$ quantifier is not allowed, it is impossible because nothing which can only match one string, can match a string longer than itself. Of course one can still invent something like `\q` which only matches `/\q/`, and still say expressions with that regular. But apparently nothing like this is supported by major implementations.

edited yesterday

[solidsnack](#)
103 3

answered Jun 16 '14 at 15:09



[jimmy23013](#)
21.3k 5 38 95

4 Impressive. I spent a while trying to get it to match something else, to no success. – [primo](#) Jun 16 '14 at 15:18

39 how (the hell) could an human produce such a thing? – [xem](#) Jun 16 '14 at 18:33

33 This deserves to be the highest voted answer on this site. – [Cruncher](#) Jun 16 '14 at 20:39

22 This is the most absurd, incredible thing I've ever seen. – [Alex A.](#) ♦ Jan 12 at 19:09

7 Someone tweeted this post so I got 49 upvotes in a day... – [jimmy23013](#) Nov 6 at 8:51
