Regex: To pull out a sub-string between two tags in a string

Asked 16 years, 4 months ago Modified 1 year, 4 months ago

Viewed 88k times

48

I have a file in the following format:

```
Data Data
Data
[Start]
Data I want
[End]
Data
```

I'd like to grab the `Data I want` from between the `[Start]` and `[End]` tags using a Regex. Can anyone show me how this might be done?

regex parsing

Share

Improve this question

Follow

edited Sep 14, 2017 at 7:01

Aditi Parikh

1,522 ● 3 ● 13 ● 34

asked Aug 4, 2008 at 13:47

Dan

29.4k ● 44 ● 151 ● 209

1 Similiar to "RegEx to get text within tags" -- stackoverflow.com/questions/353309/… – Robin Rodricks Dec 9, 2008 at 16:56

9 Answers

Sorted by: Highest score (default)

63

```
\[start\](.*?)\[end\]
```

Zhich'll put the text in the middle within a capture.

Share Improve this answer

Follow

edited Sep 18, 2017 at 13:36

Youcef LAIDANI

59.8k • 21 • 106 • 173

answered Aug 4, 2008 at 13:52

Karl Seguin

21.8k • 5 • 45 • 49

6 This still won't catch strings that have line breaks – Doug Apr 19, 2010 at 3:22

2 @Doug use option dotall. Not a problem of the regex. – AlexR Sep 12, 2014 at 8:30

23

```
\[start\]\s*(((?!\[start\]|\[end\]).)+)\s*\[end\]
```

This should hopefully drop the `[start]` and `[end]` markers as well.

Share Improve this answer

Follow

edited Sep 18, 2017 at 13:35

Youcef LAIDANI

59.8k • 21 • 106 • 173

answered Aug 4, 2008 at 13:55

Xenph Yan

84k • 16 • 50 • 55

2 The look ahead may be less efficient but I like how you prevented it from breaking if there's an unexpected `[start]` or `[end]`. It's always good to think about edge cases and preempt them. – Alex W Jul 13, 2015 at 21:08

5

```
$text ="Data Data Data start Data i want end
Data";
($content) = $text =~ m/ start (.*) end /;
print $content;
```

I had a similar problem for a while & I can tell you this method works...

Share Improve this answer

Follow

answered Oct 6, 2012 at 16:52

PhaZe

59 • 1 • 1

4

While you can use a regular expression to parse the data between opening and closing tags, you need to think long and hard as to whether this is a path you want to go down. The reason for it is the potential of tags to nest: if nesting tags could ever happen or may ever happen, the

language is said to no longer be regular, and regular expressions cease to be the proper tool for parsing it.

Many regular expression implementations, such as PCRE or perl's regular expressions, support backtracking which can be used to achieve this rough effect. But PCRE (unlike perl) doesn't support unlimited backtracking, and this can actually cause things to break in weird ways as soon as you have too many tags.

There's a very commonly cited blog post that discusses this more, http://kore-nordmann.de/blog/do_NOT_parse_using_regexp.html (google for it and check the cache currently, they seem to be having some downtime)

Share Improve this answer

Follow

answered Sep 15, 2008 at 14:18

Daniel Papasian

16.4k ● 6 ● 31 ● 32

4

A more complete discussion of the pitfalls of using a regex to find matching tags can be found at How do I find matching/nesting anything?. In particular, be aware that nesting tags really need a full-fledged parser in order to be interpreted correctly.

Note that case sensitivity will need to be turned off in order to answer the question as stated. In perl, that's the `i` modifier:

```
$ echo "Data Data Data [Start] Data i want [End]
Data" \
  | perl -ne '/\[start\](.*?)\[end\]/i; print
"$1\n"'
 Data i want
```

The other trick is to use the `?` quantifier which turns off the greediness of the captured match. For instance, if you have a non-matching `[end]` tag:

```
Data Data [Start] Data i want [End] Data [end]
```

you probably don't want to capture:

```
 Data i want [End] Data
```

Share Improve this answer

Follow

edited Aug 10, 2023 at 0:30

brian d foy

132k ● 31 ● 211 ● 604

answered Aug 20, 2008 at 19:14

Jon Ericson

21.5k ● 12 ● 102 ● 151

3

Well, if you guarantee that each start tag is followed by an end tag then the following would work.

```
\[start\](.*?)\[end\]
```

However, If you have complex text such as the follwoing:

```
[start] sometext [start] sometext2 [end] sometext
[end]
```

then you would run into problems with regex.

Now the following example will pull out all the hot links in a page:

```
'/<a(.*?)a>/i'
```

In the above case we can guarantee that there would not be any nested cases of:

```
'<a></a>'
```

So, this is a complex question and can't just be solved with a simple answer.

Share Improve this answer

Follow

answered May 11, 2009 at 20:08

Avid Coder

18.4k ● 14 ● 64 ● 68

1

With Perl you can surround the data you want with ()'s and pull it out later, perhaps other languages have a similar feature.

```
if ($s_output =~ /(data data data data START(data
data data)END (data data)/)
{
    $dataAllOfIt = $1;      # 1 full string
    $dataInMiddle = $2;     # 2 Middle Data
```

```
    $dataAtEnd = $3;        # 3 End Data
}
```

Share Improve this answer

Follow

edited Oct 12, 2008 at 12:21

brian d foy

132k ● 31 ● 211 ● 604

answered Aug 4, 2008 at 14:00

Grant

12k ● 14 ● 43 ● 48

0

Refer to this question to pull out text between tags with space characters and dots (`.`)

`[\S\s]` is the one I used

Regex to match any character including new lines

Share Improve this answer

Follow

edited May 23, 2017 at 11:46

Community Bot

1 ● 1

answered Aug 28, 2013 at 21:12

ankitkpd

673 ● 6 ● 19

0

Reading the text with in the square brackets [] i.e.[Start] and [End] and validate the array with a list of values.

jsfiddle http://jsfiddle.net/muralinarisetty/r4s4wxj4/1/

```
var mergeFields = ["[sitename]",
                   "[daystoholdquote]",
```

```
                    "[expires]",
                    "[firstname]",
                    "[lastname]",
                    "[sitephonenumber]",
                    "[hoh_firstname]",
                    "[hoh_lastname]"];

var str = "fee [sitename] [firstname] \
sdfasd [lastname] ";
var res = validateMeargeFileds(str);
console.log(res);

function validateMeargeFileds(input) {
    var re = /\[\w+]/ig;
    var isValid;
    var myArray = input.match(re);

    try{
        if (myArray.length > 0) {
            myArray.forEach(function (field) {

                isValid = isMergeField(field);

                if (!isValid){
                   throw e;
                }
            });
        }
    }
    catch(e) {
    }

    return isValid;
}
```

Share Improve this answer

Follow

edited Feb 7, 2016 at 0:38

Markus Safar

6,583 ● 5 ● 31 ● 45

answered Sep 12, 2014 at 8:26

N Murali Mohan
1

Highly active question. Earn 10 reputation (not counting the association bonus) in order to answer this question. The reputation requirement helps protect this question from spam and non-answer activity.