Using regular expressions how do I find a pattern surrounded by two other patterns without including the surrounding strings?

Asked 16 years, 2 months ago Modified 16 years, 2 months ago

Viewed 2k times

4

I want to use regular expressions (Perl compatible) to be able to find a pattern surrounded by two other patterns, but not include the strings matching the surrounding patterns in the match.

For example, I want to be able to find occurrences of strings like:

> Foo Bar Baz

But only have the match include the middle part:

> Bar

I know this is possible, but I can't remember how to do it.

regex

Share

Improve this question

Follow

edited Oct 10, 2008 at 15:07

Tanktalus

22.3k ● 5 ● 43 ● 70

asked Oct 10, 2008 at 14:53

LJ.

885 ● 2 ● 9 ● 13

4 Answers

Sorted by: Highest score (default)

Parentheses define the groupings.

```
"Foo (Bar) Baz"
```

Example

```
~> cat test.pl
$a = "The Foo Bar Baz was lass";

$a =~ m/Foo (Bar) Baz/;

print $1,"\n";
~> perl test.pl
Bar
```

Share Improve this answer

Follow

answered Oct 10, 2008 at 14:56

Vinko Vrsalovic

340k ● 55 ● 340 ● 373

> Wow you actually gave an entire script I have to vote this one up. – Cervo Oct 10, 2008 at 15:12

In the general case, you probably can't. The simplest approach is to match everything and use backreferences to capture the portion of interest:

```
Foo\s+(Bar)\s+Baz
```

This isn't the same as not including the surrounding text in the match though. That probably doesn't matter if all

you want to do is extract "Bar" but would matter if you're matching against the same string multiple times and need to continue from where the previous match left off.

Look-around will work in some cases. Tomalak's suggestion:

```
(?<=Foo\s)Bar(?=\sBaz)
```

only works for fixed width look-behind (at least in Perl). As of Perl 5.10, the `\K` assertion can be used to effectively provide variable width look-behind:

```
Foo\s+\KBar(?=\s+Baz)
```

which should be capable of doing what you asked for in all cases, but would require that you're implementing this in Perl 5.10.

While it would be convenient, there's no equivalent of `\K` for ending the matched text, so you have to use a look-ahead.

Share Improve this answer

Follow

answered Oct 10, 2008 at 20:17

Michael Carman
30.8k ● 10 ● 79 ● 124

Excellent. I'd give +2 if I could :) – Vinko Vrsalovic Oct 11, 2008 at 7:36

I'd also give it +2, because of the references to the new features in Perl 5.10. – Brad Gilbert Oct 16, 2008 at 4:27

4

Use lookaround:

```
(?<=Foo\s)Bar(?=\sBaz)
```

This would match any "Bar" that is preceded by "Foo" and followed by "Baz", separated through a single white space. "Foo" and "Baz" would not be part of the final match.

Share Improve this answer Follow

edited Oct 10, 2008 at 15:12

answered Oct 10, 2008 at 14:56

Tomalak

338k ● 68 ● 545 ● 635

this won't work...from the perl regular expression manual perlre: (?<=pattern) A zero-width positive lookbehind assertion. For example, /(?<=\t)\w+/ matches a word following a tab, without including the tab in $&. Works only for fixed-width lookbehind. Basically (?<=Foo\s*) won't work – Cervo Oct 10, 2008 at 15:01

Oops, I did not think of that. Pattern adapted. – Tomalak Oct 10, 2008 at 15:07

Your regex return the spaces before and after Bar. (?<=Foo\s)Bar(?=\sBaz) should do the job if there is only one space before and after Bar. – Julien Hoarau Oct 10, 2008 at 15:11

Done already. Clicked "post" too early, you saw an intermediate version. ;-) – Tomalak Oct 10, 2008 at 15:12

2

$string =~ m/Foo (Bar) Baz/

$1

This may not be exactly what you want as the match is still "Foo Bar Baz". But it shows you how to just get the part that you are interested in. Otherwise you can use lookahead and lookbehind to get the match without consuming characters...

Share Improve this answer

Follow

answered Oct 10, 2008 at 14:57

Cervo

3,082 ● 1 ● 25 ● 27