Perl one-liner to extract groups of characters

Asked 10 years, 6 months ago Modified 10 years, 6 months ago Viewed 13k times

15

I am trying to extract a group of characters with a Perl one-liner, but I have been unsuccessful:

```
echo "hello_95_.txt" | perl -ne 's/.*([0-9]+).*/\1/'
```

Returns nothing, while I would like it to return `95`. How can I do this with Perl?

Update:

Note that, in contrast to the suggested duplicate, I am interested in how to do this from the *command-line*. Surely this looks like a subtle difference, but it's not straightforward unless you already know how to effectively use Perl one-liners.

Since people are asking, eventually I want to learn to use Perl to write powerful one-liners, but most immediately I need a one-liner to extract consecutive digits from each line in a large text file.

perl

Share Edit Follow edited Jul 15, 2013 at 23:25 asked Jul 15, 2013 at 22:51

Amelio Vazquez-Reina
93.2k 137 363 568

1 This isn't perl, it's sed. What are you trying to do? Extract the integer from the filename? – user2579943 Jul 15, 2013 at 22:57

Thanks @YatinSaraiya I have updated my OP. I would like to use Perl, not sed. – Amelio Vazquez-Reina Jul 15, 2013 at 23:01

Eventually I would like to use this one-liner to extract consecutive digits from each line in a large text file – Amelio Vazquez-Reina Jul 15, 2013 at 23:04

Thanks @YatinSaraiya You are probably right, but I am trying to switch to Perl for most text extraction/replacement tasks (learning the syntax for sophisticated queries with awk/sed/etc can be tedious) – Amelio Vazquez-Reina Jul 15, 2013 at 23:06

1 possible duplicate of How to extract a number from a string in Perl? – djf Jul 15, 2013 at 23:07

2 Answers

Sorted by: Highest score (default)

18

```
perl -pe's/\D*(\d+).*/$1/'
```

or

```
perl -nE'/\d+/&&say$&'
```

or

```
perl -nE'say/(\d+)/'
```

or

```
perl -ple's/\D//g'
```

or may be

```
perl -nE'$,=" ";say/\d+/g'
```

Share Edit Follow

edited Jul 15, 2013 at 23:30

answered Jul 15, 2013 at 23:04
Hynek -Pichi- Vychodil
26.3k 5 53 75

`$&` better written as `(\d+) && say $1` . – squiguy Jul 15, 2013 at 23:13

@squiguy Yep, it would be faster but longer. – Hynek -Pichi- Vychodil Jul 15, 2013 at 23:17

2 Thanks! - What is the difference between these three options? – Amelio Vazquez-Reina Jul 15, 2013 at 23:22

3 Eww! The OP asked for a one-liner, not for code golf. Use some spaces. (Great answer otherwise, though, but seriously, `&&say$&` ?) – Ilmari Karonen Dec 28, 2013 at 16:27

3 For anyone else going through this hell: The single quotes apparently matter (you can't use double quotes for the command). At least on tcsh, `echo what the frack | perl -pe's/^wh(at)/$1/'` produces different output than `echo what the frack | perl -pe"s/^wh(at)/$1/"` – makhdumi Apr 16, 2015 at 22:57

1

Well first, you need to use the -p rather than the -n switch.

And you need to amend your regular expression, as in:

```
echo "hello_95_.txt" | perl -pe "s/^.*?([0-9]+).*$/\1/"
```

which looks for the longest non-greedy string of chars, followed by one or more digits, followed by any number of chars to the end of the line.

Note that while '\1' is acceptable as a back-reference and is more familiar to SED/AWK users, '$1' is the more up-to-date form. So, you might wish to use:

```
echo "hello_95_.txt" | perl -pe "s/^.*?([0-9]+).*$/$1/"
```

instead.

Share Edit Follow

edited Jul 15, 2013 at 23:16

answered Jul 15, 2013 at 23:04

Rob Raisch

17.2k 4 48 58

What the deal with the control-A in the RHS? – tchrist Jul 15, 2013 at 23:07

1 `$ echo "hello_95_.txt" | perl -Mwarnings=FATAL,all -pe "s/^.*?([0-9]+).*$/\1/" Illegal variable name. $ echo "hello_95_.txt" | perl -Mwarnings=FATAL,all -pe 's/^.*?([0-9]+).*$/\1/' \1 better written as $1 at -e line 1. Exit 255` – tchrist Jul 15, 2013 at 23:12

1 "Current" accepted form? Do you realize what you are talking about by "current"? Also, your double quotes are gebroken. – tchrist Jul 15, 2013 at 23:16

1 I beg your pardon? ***Who do you think wrote that text?*** The warning is there because you should not do it. And you still have broken double quotes. As for operating systems, you seem to have mistaken the shell for an operating system. It isn’t. – tchrist Jul 15, 2013 at 23:21

2 For the record, @tchrist is most certainly the best source of information related to Perl and I defer to his wisdom. – Rob Raisch Jul 15, 2013 at 23:25