Cheatography

Home Cheat Sheets Create Community Help

**≛** Downloads

LOGIN

REGISTER



 $\bigcirc$  46 Comments | Rating:  $\bigstar \bigstar \bigstar \bigstar (180)$  |  $\bigcirc$ 

### Regular Expressions Cheat Sheet by DaveChild

A quick reference guide for regular expressions (regex), including symbols, ranges, grouping, assertions and some sample patterns to get you started.

### Search this cheat sheet:

#### **Anchors**

- Start of string, or start of line in multiline pattern
- Start of string
- End of string, or end of line in multiline pattern
- End of string ١Z
- Word boundary \b
- Not word boundary \B
- Start of word
- End of word

#### **Character Classes**

/c	Control character
\s	White space

- \S Not white space
- Digit \d
- Not digit \D
- Word \w
- Not word \W
- Hexadecimal digit \x
- \O Octal digit

#### **Ouantifiers**

- 0 or more {3} Exactly 3 3 or more 1 or more {3,} {3,5} 3, 4 or 5
- Add a? to a quantifier to make it ungreedy.

## Want to Get Certified for OpenStack?



#### **Escape Sequences**

- Escape following character
- \Q Begin literal sequence
- End literal sequence

"Escaping" is a way of treating characters which have a special meaning in regular expressions literally, rather than as special characters.

#### **Common Metacharacters**

\$

The escape character is usually \

#### **Groups and Ranges**

- Any character except new line (\n)
- (a|b) a or b
- (...) Group
- (?:...) Passive (non-capturing) group
- [abc] Range (a or b or c)
- [^abc] Not (a or b or c)
- [a-q] Lower case letter from a to q
- [A-Q] Upper case letter from A to Q
- [0-7]Digit from 0 to 7
- \x Group/subpattern number "x"

#### Ranges are inclusive.

#### **Pattern Modifiers**

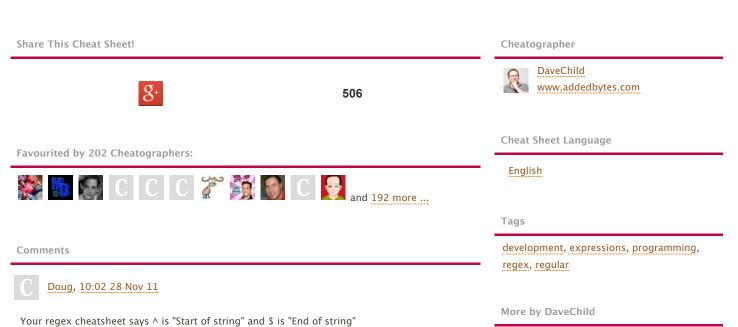
- Global match g
- i \* Case-insensitive
- Multiple lines m \*
- Treat string as single line
- Allow comments and whitespace in pattern
- e \* Evaluate replacement
- U \* Ungreedy pattern
- \* PCRE modifier

Regular Express	ions Cheat Sh	eet by DaveChild - Cheatography.co	om: Cheat Sheets	For Every Occasion
	Special	Characters	String Replacement	
Upper case letters	\n	New line	\$n	nth non-passive group
Lower case letters	\r	Carriage return	\$2	"xyz" in /^(abc(xyz))\$/
All letters	\t	Tab	\$1	"xyz" in /^(?:abc)(xyz)\$/
Digits and letters	\v	Vertical tab	\$`	Before matched string
Digits	\f	Form feed	\$'	After matched string
Hexadecimal digits	\xxx	Octal character xxx	\$+	Last matched string
Punctuation	\xhh	Hex character hh	\$&	Entire matched string
Space and tab			Some	regex implementations use \ instead of
Blank characters			\$.	
Control characters				
Printed characters				
Printed characters and spaces				
Digits, letters and underscore				
Lookahead assertion				
Negative lookahead				
Lookbehind assertion				
Negative lookbehind				
Once-only Subexpression				
Condition [if then]				
Condition [if then else]				
Comment				
	Upper case letters Lower case letters All letters Digits and letters Digits Hexadecimal digits Punctuation Space and tab Blank characters Control characters Printed characters Printed characters Digits, letters and underscore  Lookahead assertion Negative lookahead Lookbehind assertion Negative lookbehind Once-only Subexpression Condition [if then] Condition [if then else]	Upper case letters \n Lower case letters \r All letters \t Digits and letters \t Punctuation \t Space and tab Blank characters Control characters Printed characters Printed characters Digits, letters and underscore  Lookahead assertion Negative lookahead Lookbehind assertion Negative lookbehind Once-only Subexpression Condition [if then] Condition [if then else]	Upper case letters Lower case letters All letters Digits and letters Hexadecimal digits Punctuation Space and tab Blank characters Printed characters Printed characters Printed characters Printed characters Digits, letters and underscore  Lookahead assertion Negative lookahead Lookbehind Once-only Subexpression Condition [if then] Condition [if then else]	Upper case letters \ \n \ New line \ \ \square   Sn \ Lower case letters \ \ \r \ Carriage return \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \

# Download to PDF

It's Free to Download and Convert. Get In Seconds. Get ConvertDocsNow.

0 0



Linux Command Line Cheat Sheet

**CSS2 Cheat Sheet** 



DaveChild, 10:02 28 Nov 11

Hi Doug. I've clarified that section. Thanks for the heads up :)



, 10:02 28 Nov 11

It looks unchanged to me. A still says it's "start of string" and \$ still says "end of string". They should be reversed, right?



, 10:02 28 Nov 11

Actually, I'm sorry, you're right! I was pretty confused there, sorry if I've confused anyone else.



DaveChild, 10:02 28 Nov 11

No worries Keith - easy mistake to make!



Tom Hunter, 10:02 28 Nov 11

Can you tag this as 'regex'? Searching for regex on cheatography yields two other results, but not this one.



Doug, 22:52 12 Apr 12

Would you add \Q... \E to the cheatsheet?



Jorge, 12:16 4 May 12

It would be nice to see the list of white space characters



Jeff, 15:44 9 May 12

Great resource! Thanks for putting this together and sharing.



david, <u>08:58 20 May 12</u>

What language/flavor is this? Or I should also ask, if non is specified, what tends to be the default? I'm specifically looking for php or javascript, and I know they're all mostly the same, but not 100%.

Thanks!



Chris, 09:19 7 Jun 12

David, Regex is programming language neutral, as in, it doesn't matter if you are programming regex expressions in javascript, c#, c++, PHP, or even command line \*nix, makes no difference. Only thing you have to watch out for is some programming languages may require

mod\_rewrite Cheat Sheet

PHP Cheat Sheet

Wikipedia Cheat Sheet

#### Lists

This cheat sheets appears on these lists:

LAMP Cheat Sheet Collection by DaveChild

#### **Related Cheat Sheets**

This cheat sheets shares tags with:

.NET Regular Expressions by Cheatography

Google Analytics Regular Expressions by Jay Taylor

Perl Reference Card by mishin

#### Thumbnail



different various regex characters to be escaped differently (so the programming language doesn't try to interpret it). Usually a backslash. For instance \\ means ONE backslash in many languages.



Andy Grosland, 09:45 28 Jun 12

Very handy, thank you!

Don't forget Perl ;-)



littleguy, 15:23 6 Aug 12

Great and useful stuff!



Chilean, 14:03 23 Aug 12

Hi, I'm trying to learn REGEX, and I need to find this: "Page 1 Of 60", ..... "Page 50 of 60", But I can't find it using reg. expressions! :(. How would you do that? Thank you!



kris w, 08:50 13 Sep 12

Is there a cheat sheet to the cheat sheet? Is this in plain english anywhere?..."negative lookahead"..huh?



Travis, 13:07 4 Oct 12

@david, this cheat sheet is pretty neutral. The most common flavor is Perl Compatible Regular Expressions (PCRE). Javascript's engine is close to that and PHP also has Perl Compatible functions for Regex; they use the PREG prefix. Most everything on this sheet should be supported by PHP's engine (I think POSIX character classes are not). Javascript's engine isn't as featureful. Some advanced features aren't supported, but all the basics are there. If you need a multiline match and you can't use the flag, you can use an inverted class range such as [\s\S] in place of the . (dot) to match anything including newlines.

@Chilean+kris w, You need to find a resource for learning Regular Expressions. This cheat sheet is for reference, not learning. Check out http://www.regular-expressions.info/



fsnow55, 16:17 22 Oct 12

I was confused by the first comment (which was wrong, but you compounded the error with an acknowlegement). A is the start of string or line. Period.

Also, your cheat sheet is better organized than the more comprehensive http://www.regular-expressions.info/

since its more succinct. The latter has a 1-page summary but its too verbose.



ty, 12:50 22 Nov 12

Nice sheet.



Is there a reason why the ']' character is not listed under metacharacters? Doesn't that character require to be escaped if searched for?



Hey Dave. Thanks for the cheat sheet. You may want to change "Not a or b or c" when you describe the [ $^a$ bc] negated character class, because in English, the negation is ambiguous. It could mean "neither a nor be nor c." Or the "a" could be the only negated disjunct. You could mean ( $^a$  v (b v c)).



Could be added to the list.

Case Conversion

\I Make next character lowercase

\u Make next character uppercase

\L Make entire string (up to \E) lowercase

\U Make entire string (up to \E) uppercase

\u\L Capitalize first char, lowercase rest (sentence)



I have a database using regex. I am trying to use ^file to get all files with name file\_,file ,file\_name\_date. but it is not working any help

david.baird, 18:59 15 Jul 13

Is \x supported anywhere? I can't find examples of it in use searching the web. It also does not work in a script on my Macintosh, OSX 10.7.5 using the OS's perl installation.

Drew White, <u>15:10 13 Feb 14</u>

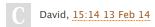
I'm trying to come up with a regex string to filter results to a directory that includes a-zA-Z but that also includes an underscore ('\_'). Do you know of a way to do this?

David, <u>15:14 13 Feb 14</u>

Is \x supported anywhere? I can't seem to find where it is supported. E.g., perl on MacOS 10.7.5. Searching for on-line examples or help also fails, in that no one knows about it.

David, <u>15:14 13 Feb 14</u>

Is  $\x (Regular Expressions Character Classes)$  supported anywhere? Is this a new class that has just been added, because I am unable to use it in working with IPV6 addresses. Searching for a string containing something like 2001::1a79 with a RegEx 2001:: $\x \{1,4\}$  will fail, but if I use 2001:: $\arrow [a-fA-F0-9]\{1,4\}$  will work.



\x is a term in "Regular Expressions Character Classes" for an hexadecimal digit. How does this compare to the \xhh "Special Characters"? Is it supported today?



It would be great to increase in some ways the --> : <---- in the

"(?:...) Passive (non-cÂaptÂuring) group" description. Just after the "?", it is praticaly not visible. Thanks!

Simon, <u>17:33 17 Feb 14</u>

This is a great cheat-sheet. Two minor niggles:

- \* Would be great to hint on the characters hidden in the character classes (\s = [ \t\n\r\f], \d = [0-9], \w = [a-zA-Z\_0-9])
- \* I think possibly there's a mistake in the section "Special Characters"  $\xx$  is probably not the octal character xxx. See "Character Classes": it should be  $\ax{O}$  (and by the way: why are  $\ax{O}$  and  $\xx$  duplicated in "Special Characters" and "Character Classes"...)

And I support Edir's request for a section "Case Conversion".

But again: great sheet, thanks!

Sahana A V, <u>17:48 17 Feb 14</u>

Thank you for the Regex cheat sheet :-)

Mervin, <u>13:37 19 Apr 14</u>

Thank you. It would have been better if you would have included test, match, etc.

Bhaggs, <u>11:33 21 Apr 14</u>

Does these support ereg too?

yashawanth, <u>10:24 17 May 14</u>

what is mean by (.\*?) please let me know as soon as possible ??

Roedy Green, 20:57 26 May 14

There are so many dialects of regex. The most important fact should be right up top, which dialects do you cover?

Ted, 16:34 13 Jun 14

Can you please fix the pdf so it is able to download? Right now my browser (Google Chrome)

only show a raw pdf instead of downloading it. Thank you very much :)



I agree with Roedy Green. I came here looking for specific a version of regex. This is still so helpful.



Ted, when the PDF displays in Chrome, right-click in the page and select Save As.

jaya prakash, <u>09:35 31 Oct 14</u>

Hi,

Can u help me to find regular expression --

SELECT distinct col\_1 FROM tablename WHERE (SUBSTR(col\_1,-1,1)) = '5'

need resusIt as abc-cxy-5

not as abc-cxy-65 not as abc-cxy-555 not as abc-cxy-05545

Ken Sanders, <u>09:12 5 Nov 14</u>

Hello Jaya,
MySQL supports regular expressions:
select distinct col\_1
from tablename
where col\_1
REGEXP '[a-z]{3}-[a-z]{3}-5'

Jeff, <u>08:50 11 Mar 15</u>

Great Cheatsheet. Maybe you could add the toggles like (?i...), (?-i...), (?i: ...), (?-i: ...) and their cousins with "m" and "x".

THanks for the great work

Hemant Bellani, <u>12:26 29 Mar 15</u>

Thank you for this !!

TME520, 03:31 23 Apr 15

Nice cheat sheet, great job!



Sudhakar, 18:36 25 May 15

Hi Team,

How to write a regular expression for this kind of below line present in document .

Ref: 000/SP/00-000

I need to set Target data formats and Keywords for this field.

Thanks in advance.



ms\_jo553698, 17:25 20 Jun 15

I have been searching for programs that others might be using to roll dice in Pachisi on the internet. I know nothing about programming and don't know how to word the question. Are there cheat sheets out there for something like this?



matu, 17:45 29 Jun 15

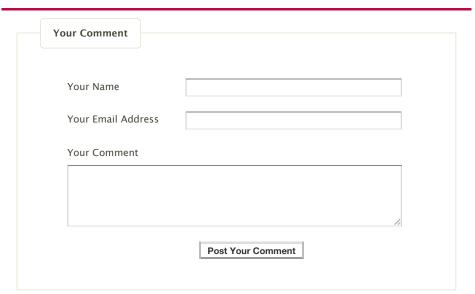
If you like this you may want to check out my tutorial on how to use regular expressions in Ruby :) http://www.blackbytes.info/2015/06/mastering-ruby-regex/



romeoh, 08:56 5 Aug 15

good

Add a Comment



9/1/2015

Latest Cheat Sheet

**Most Popular Cheat Sheet** 

**Random Cheat Sheet** 

\*\*\*\*

2nd May

Pasos para aumentar tu influencia sobre los demás Cheat Sheet

**Regular Expressions Cheat Sheet** 

ServiceNow Syntax Macros Cheat Sheet

#### **About Cheatography**

<u>Cheatography</u> is a collection of 1268 cheat sheets and quick references in 20 languages for everything from google to maths!

#### On The Blog

Cheat Sheet Statistics with Google Analytics!

Want to know more about the people downloading your cheat sheets? No problem! You can now link your cheat sheets to your Google Analytics account!

#### **Recent Activity**



oli.p updated Eclipse.
5 hours 39 mins ago



AdrianArroyoCalle published Pasos para aumentar tu influencia sobre los demás.

12 hours 27 mins ago

mdoehle updated PKI / openSSL.

14 hours 33 mins ago



**Behind the Scenes** 

If you have any problems, or just want to say hi, you can find us right here:



DaveChild



SpaceDuck

**Cheatography RSS Feeds** 

Latest Cheat Sheets Latest Blog Posts Share Cheatography!

1.3K

© 2011 - 2015 Cheatography.com | CC License | Terms | Privacy

Site by Added Bytes Magento Development Brighton