

# Regular Expressions Cheat Sheet by DaveChild

0

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

[development](#) [regex](#) [regular](#) [expressions](#) [programming](#)

## Anchors

<code>^</code>	Start of string, or start of line in multi-line pattern
<code>\A</code>	Start of string
<code>\$</code>	End of string, or end of line in multi-line pattern
<code>\Z</code>	End of string
<code>\b</code>	Word boundary
<code>\B</code>	Not word boundary
<code>\&lt;</code>	Start of word
<code>\&gt;</code>	End of word

## Character Classes

<code>\c</code>	Control character
<code>\s</code>	White space
<code>\S</code>	Not white space
<code>\d</code>	Digit
<code>\D</code>	Not digit
<code>\w</code>	Word
<code>\W</code>	Not word
<code>\x</code>	Hexadecimal digit
<code>\O</code>	Octal digit

## Quantifiers

<code>*</code>	0 or more	<code>{3}</code>	Exactly 3
<code>+</code>	1 or more	<code>{3,}</code>	3 or more
<code>?</code>	0 or 1	<code>{3,5}</code>	3, 4 or 5

Add a `?` to a quantifier to make it ungreedy.

## Escape Sequences

<code>\</code>	Escape following character
<code>\Q</code>	Begin literal sequence
<code>\E</code>	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

<code>^</code>	<code>[</code>	<code>.</code>	<code>\$</code>
<code>{</code>	<code>*</code>	<code>(</code>	<code>\</code>
<code>+</code>	<code>)</code>	<code> </code>	<code>?</code>
<code>&lt;</code>	<code>&gt;</code>		

The escape character is usually `\`

## Special Characters

<code>\n</code>	New line
<code>\r</code>	Carriage return
<code>\t</code>	Tab
<code>\v</code>	Vertical tab
<code>\f</code>	Form feed
<code>\xxx</code>	Octal character xxx
<code>\xhh</code>	Hex character hh

## Groups and Ranges

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

Ranges are inclusive.

## Pattern Modifiers

<code>g</code>	Global match
<code>i *</code>	Case-insensitive
<code>m *</code>	Multiple lines
<code>s *</code>	Treat string as single line
<code>x *</code>	Allow comments and whitespace in pattern
<code>e *</code>	Evaluate replacement
<code>U *</code>	Ungreedy pattern

\* PCRE modifier

[upper:]	Upper case letters
[lower:]	Lower case letters
[alpha:]	All letters
[alnum:]	Digits and letters
[digit:]	Digits
[xdigit:]	Hexadecimal digits
[punct:]	Punctuation
[blank:]	Space and tab
[space:]	Blank characters
[cntrl:]	Control characters
[graph:]	Printed characters
[print:]	Printed characters and spaces
[word:]	Digits, letters and underscore

\$n	nth non-passive group
\$2	"xyz" in /^(abc(xyz))\$/
\$1	"xyz" in /^(?:abc)(xyz)\$/
\$`	Before matched string
\$'	After matched string
\$+	Last matched string
\$&	Entire matched string

Some regex implementations use \ instead of \$.

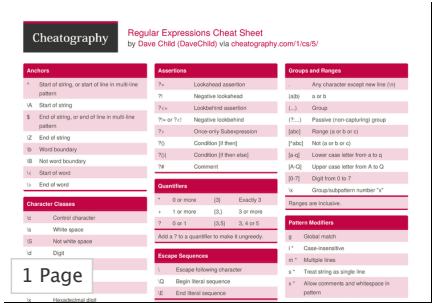
Assertions

?=	Lookahead assertion
?!	Negative lookahead
?<=	Lookbehind assertion
?!= or ?<!	Negative lookbehind
?>	Once-only Subexpression
?()	Condition [if then]
?()	Condition [if then else]
?#	Comment


Download the Regular Expressions Cheat Sheet

Thank The Author!

You can use [Flattr](#) to thank the author of this cheat sheet!



PDF (recommended)

 PDF (1 page)

Alternative Downloads

 PDF (black and white)

 LaTeX



Like this cheat sheet? Check out our sponsors!



writing better. More readable content means **higher conversion rates** and **better reader engagement**. Measure website and document readability, measure keyword density and more!

[Click Here To Get Started!](#)

## Comments

Doug, [10:02 28 Nov 11](#)

Your regex cheatsheet says ^ is "Start of string" and \$ is "End of string"

[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. ^ 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](#)

## Cheatographer

DaveChild

[www.getpostcookie.com](http://www.getpostcookie.com)

## Metadata

Languages: [English](#)

Published: 19th October, 2011

Rated: 5 stars based on 238 ratings

## Lists

[LAMP Cheat Sheet Collection](#) by DaveChild

## Favourited By

and [307 more ...](#)

Jeff, [15:44 9 May 12](#)

Great resource! Thanks for putting this together and sharing.

david, [08:58 20 May 12](#)

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 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

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 acknowledgement). ^ 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.

Rob, [11:47 24 Jan 13](#)

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

Bill, [11:48 24 Jan 13](#)

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

Edir, [09:17 15 Feb 13](#)

Could be added to the list.

#### Case Conversion

\l 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)

Gabe, [17:16 26 Mar 13](#)

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.

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, [15:14 13 Feb 14](#)

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, [15:14 13 Feb 14](#)

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::[a-fA-F0-9]{1,4} will work.

David, [15:14 13 Feb 14](#)

\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?

Benoit, [08:54 14 Feb 14](#)

It would be great to increase in some ways the --> : <---- in the "(?:...) Passive (non-capturing) group" description. Just after the "?", it is practically not visible. Thanks!

Simon, [17:33 17 Feb 14](#)

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" - \xxx is probably not the octal character xxx. See "Character Classes": it should be \Oxxx (and by the way: why are \O and \x 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, [17:48 17 Feb 14](#)

Thank you for the Regex cheat sheet :-)

Mervin, [13:37 19 Apr 14](#)

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

Bhaggs, [11:33 21 Apr 14](#)

yashawanth, [10:24 17 May 14](#)

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 :)

William, [09:21 10 Jul 14](#)

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

Aaron, [11:13 8 Sep 14](#)

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

jaya prakash, [09:35 31 Oct 14](#)

Hi ,  
Can u help me to find regular expression --

```
SELECT distinct col_1 FROM tablename  
WHERE (SUBSTR(col_1,-1,1)) = '5'
```

need resuslt as abc-cxy-5

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

Ken Sanders, [09:12 5 Nov 14](#)

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

Jeff, [08:50 11 Mar 15](#)

cousins with m and a .

THanks for the great work

Hemant Bellani, [12:26 29 Mar 15](#)

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

Shamim, [09:11 14 Sep 15](#)

Thanks for the great job!

Prabhakaran Govindaraj, [10:59 7 Oct 15](#)

Thanks for the nice and comprehensive resource.



Hi I am a techno retard I gather Regex is coding flavour. However I have no idea what you write is there sny resources.

Sorry for stupidity.

Philbo.

DZanke, 14:16 28 Nov 15

Great list! I'm new to Teradata Regular Expressions and couldn't find them anywhere.

Ish, 21:27 26 Jan 16

Very helpful. Thanks.

Shrirang Garge, 13:29 24 Feb 16

I am finding it difficult to write a regex for the date input..it looks like this...[31-Mar-2015:06:22:48 -600]. I am trying with [0-9a-zA-Z] but giving me null values. Please help.

pradeep, 13:29 24 Feb 16

I need to split a group of element baased on "," delimiter this is the example  
input :"(10,{10,9,8,7,6,5,4,3,2,1}),(8,{8,7,6,5,4,3,2,1}),(8,{8,7,6,5,4,3,2,1})  
output: (10,{10,9,8,7,6,5,4,3,2,1})  
(8,{8,7,6,5,4,3,2,1})  
(8,{8,7,6,5,4,3,2,1}) regex: [()], which is not getting me in the way i want.

brent, 15:28 5 Mar 16

Do you have any issue with us distributing the PDF regex cheat sheet as is with no changes to others. Before I put it on our internal collaboration tool I need to make sure there are no issues from you in doing so. I will not be modifying the PDF or removing your details from the sheet, it will be just as it is but shareable from within our company's portal.

DaveChild, 15:31 5 Mar 16

It's released (as all cheat sheets here are) under a CC license, so you can redistribute it according to the terms here:

<http://creativecommons.org/licenses/by-nc-sa/2.0/uk/>

So, go for it :)

Bartleby, 15:29 5 Mar 16

Shrirang: What're you trying to do with the date? Just capture it?

experienced people it's probably really elegant, but it's functional, anyway.

```
(\D\d{2})(\-)([A-Z]+)(\-)(\d{4})(\:\d{2}\:\d{2})(\s+)(\-\d+)(\|)
```

Bartleby, [15:29 5 Mar 16](#)

Pradeep: I was able to accomplish what you're looking for with the following text (I'm using Notepad++):

Search: (\))(.)

Replace: \1\r\n\2

tasjaevan, [08:24 26 Mar 16](#)

Hi Dave – could you an entry for free whitespace regexes using the ?x syntax?

Abdel Maghraby, [17:19 28 Mar 16](#)

Hello Guys,

I am trying to create a code to prevent white spaces before or after a string.

[ name ]

[ name]

[name ]

Ideally I want this to be strictly

[Name]

How can i achieve that?

Culbin, [17:00 12 Jul 16](#)

Hi Folks! I use regexp syntax to schedule TV recordings via TVHeadend which is case-insensitive by default.

Here's an extreme example which highlights the problem.

Simply using 'big brother' produces many different program name matches like:

Big Brother

Big Brother's Bit On The Side

Big Brother: Annihilation

Big Brother: Eviction Massacre

New: Big Brother

New: Big Brother's Bit On The Side

New: Big Brother: Live...

I'd like to tell it to ignore the Bit On The Side programs but match the rest. I've researched till I'm blue in the face with no luck.

My tiny brain tells me that in regular English it would read like this:

match 'big brother' but not if contains 'bit on the side'

but I'm lost when it comes to translating this to regexp syntax.

I honestly don't know if it accepts Lookahead or Lookbehind which I see is mentioned a lot, sorry.



## Latest Cheat Sheet

## Most Popular Cheat Sheet

★★★★★

## Random Cheat Sheet

★★★★☆

[Linux command line Cheat Sheet](#)[Regular Expressions Cheat Sheet](#)[Sublime Text 2 OSX Keyboard Shortcuts](#)

## About Cheatography

[Cheatography](#) is a collection of [2846 cheat sheets](#) and quick references in [25 languages](#) for everything from [linux](#) to [business](#)!

## On The Blog

*10th June*

### 5 Ways Cheatography Benefits Your Business

Cheatography Cheat Sheets are a great timesaver for individuals – coders, gardeners, musicians, everybody! But businesses can benefit from them as well – read on to find out more.

## Recent Activity

[alessandro.grassi](#) published [Linux command line](#).

1 hour 7 mins ago

[apowers313](#) replied to: [Latex or similar?](#)

1 day 10 hours ago

[Davidpol](#) published [60-Second Poster Evaluation](#).

3 days 8 hours ago

## Behind the Scenes

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

[DaveChild](#)[SpaceDuck](#)[Cheatography](#)

## Cheatography RSS Feeds

[Latest Cheat Sheets](#)[Latest Blog Posts](#)

## Share Cheatography!

**2.5K**