FREEFORMATTER.COM HTTPS Contact Like 1.9K

Formatters

JSON Formatter
HTML Formatter
XML Formatter
SQL Formatter
Batch Formatter (new!)

Validators

JSON Validator
HTML Validator
XML Validator - XSD
XPath Tester
Credit Card Number Generator & Validator
Regular Expression Tester
Java Regular Expression Tester
Cron Expression Generator - Quartz

Encoders & Decoders

Url Encoder & Decoder
Base 64 Encoder & Decoder
QR Code Generator

Code Minifiers / Beautifier

JavaScript Beautifier
CSS Beautifier
JavaScript Minifier
CSS Minifier

Converters

XSD Generator
XSLT (XSL Transformer)
XML to JSON Converter
JSON to XML Converter
CSV to XML Converter
CSV to JSON Converter
Epoch Timestamp To Date

Cryptography & Security

Message Digester (MD5, SHA-256, SHA-512)
HMAC Generator
MD5 Generator
SHA-256 Generator
SHA-512 Generator

String Escaper & Utilities

String Utilities
HTML Escape
XML Escape
Java and .Net Escape
JavaScript Escape
JSON Escape
CSV Escape
SQL Escape

Web Resources

Lorem Ipsum Generator
LESS Compiler
List of MIME types
HTML Entities
Url Parser / Query String Splitter
i18n - Formatting standards & code snippets
ISO country list - HTML select snippet
USA state list - HTML select snippet
Canada province list - HTML select snippet
Mexico state list - HTML select snippet
Time zone list - HTML select snippet

Java Regular Expression Tester

This **free Java regular expression tester** lets you test your regular expressions against any entry of your choice and clearly highlights all matches. It is based on the Pattern class of Java 8.0.

Consult the regular expression documentation or the regular expression solutions to common problems section of this page for examples. If you need more examples or solutions, please contact me.

Java Regular Expression :

Entry to test against :

Replace with (Optional):

You can make use of $1, $2, $3 and so on if you are using parenthesis groups in your regular expression. \t \n \r are supported.

Flags:

□ Dotall □ Comments
□ Multiline □ Literal
□ Unix lines □ Unicode case
□ Canon EQ □ Unicode character class

TEST MATCH REPLACE FIRST REPLACE ALL

Regular Expression - Documentation

Metacharacters

Character	What does it do?
$	Matches the **end of the input**. If in multiline mode, it also matches **before a line break character**, hence every end of line.
(?:x)	Matches 'x' but **does NOT remember the match**. Also known as NON-capturing parenthesis.
(x)	Matches 'x' and **remembers the match**. Also known as capturing parenthesis.
*	Matches the preceding character **0 or more times**.
+	Matches the preceding character **1 or more times**.
.	Matches **any single character except the newline character**.
?	• Matches the preceding character **0 or 1 time**. • When used after the quantifiers *, +, ? or {}, **makes the quantifier non-greedy**; it will match the minimum number of times as opposed to matching the maximum number of times.
[\b]	Matches a **backspace**.
[^abc]	Matches **anything NOT enclosed by the brackets**. Also known as a negative character set.
[abc]	Matches **any of the enclosed characters**. Also known as a character set. You can create range of characters using the hyphen character such as A-Z (A to Z). Note that in character sets, special characters (., *, +) do not have any special meaning.

\	• Used to indicate that the **next character should NOT be interpreted literally**. For example, the character 'w' by itself will be interpreted as 'match the character w', but using '\w' signifies 'match an alpha-numeric character including underscore'. • Used to indicate that a **metacharacter is to be interpreted literally**. For example, the '.' metacharacter means 'match any single character but a new line', but if we would rather match a dot character instead, we would use '\.'.
\0	Matches a **NULL character**.
\b	Matches a **word boundary**. Boundaries are determined when a word character is NOT followed or NOT preceeded with another word character.
\B	Matches a **NON-word boundary**. Boundaries are determined when two adjacent characters are word characters OR non-word characters.
\cX	Matches a **control character**. X must be between A to Z inclusive.
\d	Matches a **digit character**. Same as [0-9] or [0123456789].
\D	Matches a **NON-digit character**. Same as [^0-9] or [^0123456789].
\f	Matches a **form feed**.
\n	Matches a **line feed**.
\r	Matches a **carriage return**.
\s	Matches a **single white space character**. This includes space, tab, form feed and line feed.
\S	Matches **anything OTHER than a single white space character**. Anything other than space, tab, form feed and line feed.
\t	Matches **a tab**.
\uhhhh	Matches a character with the **4-digits hexadecimal code**.
\v	Matches **a vertical tab**.
\w	Matches **any alphanumeric character incuding underscore**. Equivalent to [A-Za-z0-9_].
\W	Matches **anything OTHER than an alphanumeric character incuding underscore**. Equivalent to [^A-Za-z0-9_].
\x	A back reference to the substring matched by the x parenthetical expression. x is a positive integer.
\xhh	Matches a character with the **2-digits hexadecimal code**.
^	• Matches the **beginning of the input**. If in multiline mode, it also matches **after a line break character**, hence every new line. • When used in a set pattern ([^abc]), it negates the set; **match anything not enclosed in the brackets**
x(?!y)	Matches 'x' **only if 'x' is NOT followed by 'y'**. Also known as a negative lookahead.
x(?=y)	Matches 'x' **only if 'x' is followed by 'y'**. Also known as a lookahead.
x|y	Matches **'x' OR 'y'**.
{n,m}	Matches the preceding character **at least n times and at most m times. n and m can be omitted if zero.**.
{n}	Matches the preceding character **exactly n times**.

Regular Expression - Solutions to common problems (Recipes)

How can I emulate DOTALL in JavaScript?

DOTALL is a flag in most recent regex libraries that makes the . metacharacter match anything INCLUDING line breaks. JavaScript by default does not support this since the . metacharacter matches anything BUT line breaks. To emulate this