Characters

Character	Legend	Example	Sample Match
\d	Most engines: one digit from 0 to 9	file_\d\d	file_25
\d	.NET, Python 3: one Unicode digit in any script	file_\d\d	file_9৪
\w	Most engines: "word character": ASCII letter, digit or underscore	\w-\w\w\w	A-b_1
\w	.Python 3: "word character": Unicode letter, ideogram, digit, or underscore	\w-\w\w\w	字-ま_۳
\w	.NET: "word character": Unicode letter, ideogram, digit, or connector	\w-\w\w\w	字-ま‿۳
\s	Most engines: "whitespace character": space, tab, newline, carriage return, vertical tab	a\sb\sc	a b c
\s	.NET, Python 3, JavaScript: "whitespace character": any Unicode separator	a\sb\sc	a b c
\D	One character that is not a*digit* as defined by your engine's *\d*	\D\D\D	ABC
\W	One character that is not a*word character* as defined by your engine's *\w*	\W\W\W\W\W	*-+=)
\S	One character that is not a*whitespace character* as defined by your engine's *\s*	\S\S\S\S	Yoyo

More Characters

Character	Legend	Example	Sample Match
.	Any character except line break	a.c	abc
.	Any character except line break	.*	whatever, man.
\.	A period (special character: needs to be escaped by a \)	a\.c	a.c
\	Escapes a special character	\.*\+\? \$\^\/\\	.*+? $^/\
\	Escapes a special character	\[\{\(\)\}\]	[{()}]

Quantifiers

Quantifier	Legend	Example	Sample Match
+	One or more	Version \w-\w+	Version A-b1_1
{3}	Exactly three times	\D{3}	ABC
{2,4}	Two to four times	\d{2,4}	156
{3,}	Three or more times	\w{3,}	regex_tutorial
*	Zero or more times	A*B*C*	AAACC
?	Once or none	plurals?	plural

More Quantifiers

Quantifier	Legend	Example	Sample Match
+	The + (one or more) is "greedy"	\d+	12345
?	Makes quantifiers "lazy"	\d+?	1 in **1**2345
*	The * (zero or more) is "greedy"	A*	AAA
?	Makes quantifiers "lazy"	A*?	empty in AAA
{2,4}	Two to four times, "greedy"	\w{2,4}	abcd
?	Makes quantifiers "lazy"	\w{2,4}?	ab in **ab**cd

More White-Space

Character	Legend	Example	Sample Match
\t	Tab	T\t\w{2}	T ab
\r	Carriage return character	see below	
\n	Line feed character	see below	
\r\n	Line separator on Windows	AB\r\nCD	AB CD
\N	Perl, PCRE (C, PHP, R...): one character that is not a line break	\N+	ABC
\h	Perl, PCRE (C, PHP, R...), Java: one horizontal whitespace character: tab or Unicode space separator		
\H	One character that is not a horizontal whitespace		
\v	.NET, JavaScript, Python, Ruby: vertical tab		
\v	Perl, PCRE (C, PHP, R...), Java: one vertical whitespace character: line feed, carriage return, vertical tab, form feed, paragraph or line separator		
\V	Perl, PCRE (C, PHP, R...), Java: any character that is not a vertical whitespace		
\R	Perl, PCRE (C, PHP, R...), Java: one line break (carriage return + line feed pair, and all the characters matched by \v)		

Character Classes

Character	Legend	Example	Sample Match
[…]	One of the characters in the brackets	[AEIOU]	One uppercase vowel
[…]	One of the characters in the brackets	T[ao]p	*Tap* or *Top*
-	Range indicator	[a-z]	One lowercase letter
[x-y]	One of the characters in the range from x to y	[A-Z]+	GREAT
[…]	One of the characters in the brackets	[AB1-5w-z]	One of either: A,B,1,2,3,4,5,w,x,y,z
[x-y]	One of the characters in the range from x to y	[-~]+	Characters in the printable section of the ASCII table.
[^x]	One character that is not x	[^a-z]{3}	A1!
[^x-y]	One of the characters **not** in the range from x to y	[^ -~]+	Characters that are**not** in the printable section of the ASCII table.
[\d\D]	One character that is a digit or a non-digit	[\d\D]+	Any characters, inc- luding new lines, which the regular dot doesn't match
[\x41]	Matches the character at hexadecimal position 41 in the ASCII table, i.e. A	[\x41-\x45]{3}	ABE

POSIX Classes

Character	Legend	Example	Sample Match
[:alpha:]	PCRE (C, PHP, R…): ASCII letters A-Z and a-z	[8[:alpha:]]+	WellDone88
[:alpha:]	Ruby 2: Unicode letter or ideogram	[[:alpha:]\d]+	кошка99
[:alnum:]	PCRE (C, PHP, R…): ASCII digits and letters A-Z and a-z	[[:alnum:]]{10}	ABCDE12345
[:alnum:]	Ruby 2: Unicode digit, letter or ideogram	[[:alnum:]]{10}	кошка90210
[:punct:]	PCRE (C, PHP, R…): ASCII punctuation mark	[[:punct:]]+	?!.,:;
[:punct:]	Ruby: Unicode punctuation mark	[[:punct:]]+	⁉,:‸\]

Anchors and Boundaries

Anchor	Legend	Example	Sample Match
^	Start of string or start of linedepending on multiline mode. (But when [^inside brackets], it means "not")	^abc .*	abc (line start)
$	End of string or end of linedepending on multiline mode. Many engine-dependent subtleties.	.*? the end$	this is the end
\A	Beginning of string (all major engines except JS)	\Aabc[\d\D]*	abc (string... ...start)
\z	Very end of the string Not available in Python and JS	the end\z	this is...\n...**the end**
\Z	End of string or (except Python) before final line break Not available in JS	the end\Z	this is...\n...**the end**\n
\G	Beginning of String or End of Previous Match .NET, Java, PCRE (C, PHP, R...), Perl, Ruby		
\b	Word boundary Most engines: position where one side only is an ASCII letter, digit or underscore	Bob.*\bcat\b	Bob ate the cat
\b	Word boundary .NET, Java, Python 3, Ruby: position where one side only is a Unicode letter, digit or underscore	Bob.*\b\кошка\b	Bob ate the кошка
\B	Not a word boundary	c.*\Bcat\B.*	copycats

Logic

Logic	Legend	Example	Sample Match
|	Alternation / OR operand	22|33	33
(...)	Capturing group	A(nt|pple)	Apple (captures "pple")
\1	Contents of Group 1	r(\w)g\1x	regex
\2	Contents of Group 2	(\d\d)\+ (\d\d)=\2\+\1	12+65=65+12
(?: ...)	Non-capturing group	A(?:nt|pple)	Apple

Inline Modifiers

None of these are supported in JavaScript. In Ruby, beware of (?s) and (?m).

Modifier	Legend	Example	Sample Match
(?i)	Case-insensitive mode (except JavaScript)	(?i)Monday	monDAY
(?s)	DOTALL mode (except JS and Ruby). The dot (.) matches new line characters (\r\n). Also known as "single-line mode" because the dot treats the entire input as a single line	(?s)From A.*to Z	From A to Z
(?m)	Multiline mode (except Ruby and JS) ^ and $ match at the beginning and end of every line	(? m)1\r\n^2$\r\n^3 $	1 2 3
(?m)	In Ruby: the same as (?s) in other engines, i.e. DOTALL mode, i.e. dot matches line breaks	(?m)From A.*to Z	From A to Z
(?x)	Free-Spacing Mode mode (except JavaScript). Also known as comment mode or whitespace mode	(?x) # this is a # comment abc # write on multiple # lines [ ]d # spaces must be # in brackets	abc d
(?n)	.NET: named capture only	Turns all (parentheses) into non-capture groups. To capture, use named groups.	
(?d)	Java: Unix linebreaks only	The dot and the ^ and $ anchors are only affected by \n	

Lookarounds

Lookaround	Legend	Example	Sample Match
(?=...)	Positive lookahead	(?=\d{10})\d{5}	01234 in**01234**56789
(?<=...)	Positive lookbehind	(?<=\d)cat	cat in 1**cat**
(?!...)	Negative lookahead	(?!theatre)the\w+	theme
(?<!...)	Negative lookbehind	\w{3}(?<! mon)ster	Munster

Character Class Operations

Class Operation	Legend	Example	Sample Match
[...-[...]]	.NET: character class subtraction. One character that is in those on the left, but not in the subtracted class.	[a-z-[aeiou]]	Any lowercase consonant
[...-[...]]	.NET: character class subtraction.	[\p{IsArabic}-[\D]]	An Arabic character that is not a non-digit, i.e., an Arabic digit
[...&&[...]]	Java, Ruby 2+: character class intersection. One character that is both in those on the left and in the && class.	[\S&&[\D]]	An non-whitespace character that is a non-digit.
[...&&[...]]	Java, Ruby 2+: character class intersection.	[\S&&[\D]&&[^a-zA-Z]]	An non-whitespace character that a non-digit and not a letter.
[...&&[^...]]	Java, Ruby 2+: character class subtraction is obtained by intersecting a class with a negated class	[a-z&&[^aeiou]]	An English lowercase letter that is not a vowel.
[...&&[^...]]	Java, Ruby 2+: character class subtraction	[\p{InArabic}&&[^\p{L}\p{N}]]	An Arabic character that is not a letter or a number

Other Syntax

Syntax	Legend	Example	Sample Match
\K	Keep Out Perl, PCRE (C, PHP, R...), Python's alternate *regex*engine, Ruby 2+: drop everything that was matched so far from the overall match to be returned	prefix\K\d+	12
\Q...\E	Perl, PCRE (C, PHP, R...), Java: treat anything between the delimiters as a literal string. Useful to escape metacharacters.	\Q(C++ ?)\E	(C++ ?)