

## Reguläre Ausdrücke (*RegEx*) - Kurzübersicht

.	Standardmäßig stimmt ein Punkt mit einem einzelnen Zeichen überein, das nicht zu einer Zeilenumbruchssequenz (`r`n) gehört. Dieses Verhalten kann aber mit den Optionen DotAll (s), LF (`n), CR (`r), `a oder (*ANYCRLF) geändert werden. Zum Beispiel würde ab. mit abc und abz und ab_ übereinstimmen.
*	Ein Sternchen findet 0 oder mehrere Vorkommen des voranstehenden Elements (Zeichen, Klasse oder Teilausdruck). Zum Beispiel würde a* mit ab und aaab übereinstimmen. Dieser reguläre Ausdruck würde auch mit dem Anfang einer beliebigen Zeichenkette ohne "a" übereinstimmen. <b>Platzhalter:</b> Das Punkt-Sternchen-Muster .* ist eines der tolerantesten Muster, weil es 0 oder mehrere Vorkommen eines <i>beliebigen</i> Zeichens finden kann (außer Zeilenumbruch: `r und `n). Zum Beispiel würde abc.*123 mit abcAnything123 sowie mit abc123 übereinstimmen.
?	Ein Fragezeichen findet 0 oder 1 Vorkommen des voranstehenden Elements (Zeichen, <u>Klasse</u> oder <u>Teilausdruck</u> ). Oder auch: "Das voranstehende Element ist optional". Zum Beispiel würde colou?r sowohl mit color als auch mit colour übereinstimmen, weil das "u" optional ist.
+	Ein Pluszeichen findet 1 oder mehrere Vorkommen des voranstehenden Elements (Zeichen, <u>Klasse</u> oder <u>Teilausdruck</u> ). Zum Beispiel würde a+ mit ab und aaab übereinstimmen. Aber im Gegensatz zu a* und a? würde das Muster a+ nicht mit dem Anfang einer Zeichenkette ohne "a" übereinstimmen.
{min,max}	Findet <i>min</i> bis <i>max</i> Vorkommen des voranstehenden Elements (Zeichen, <u>Klasse</u> oder <u>Teilausdruck</u> ). Zum Beispiel würde a{1,2} mit ab übereinstimmen, als auch die ersten zwei a's mit aaab.  Zudem würde {3} genau 3 Vorkommen finden, und {3,} 3 oder mehrere Vorkommen. Hinweis: Die angegebenen Zahlen müssen kleiner als 65536 sein. Außerdem muss der erste Wert entweder kleiner oder genauso groß wie der zweite Wert sein.
[...]	<b>Zeichenklassen:</b> Die eckigen Klammern umschließen eine Liste oder einen Bereich von Zeichen (oder beides). Zum Beispiel bedeutet [abc]: "jedes Zeichen, das entweder ein a, b oder c ist". Mit einem Bindestrich kann ein Bereich dargestellt werden; zum Beispiel bedeutet [a-z]: "jedes Zeichen von a bis z". Listen und Bereiche können kombiniert werden; zum Beispiel bedeutet [a-zA-Z0-9_]: "jedes Zeichen, das alphanumerisch oder ein Unterstrich ist".  Nach einer Zeichenklasse kann *, ?, + oder {min,max} erfolgen. [0-9]+ würde beispielsweise 1 oder mehrere Vorkommen von beliebigen Ziffern finden; somit würde es mit xyz123 übereinstimmen, aber nicht mit abcxyz.  Desweiteren kann man benannte Zeichenbereiche (POSIX) in Form von [[:xxx:]] angeben; xxx ist dabei eines der folgenden Wörter: alnum, alpha, ascii (0-127), blank (Leerzeichen oder Tabulator), cntrl (Steuerzeichen), digit (0-9), xdigit (hexadezimale Ziffern), print, graph (print ohne space), punct, lower, upper, space (Leerraum), word (das gleiche wie <u>\w</u> ).  Innerhalb einer Zeichenklasse müssen die Zeichen nicht mit einem Escapezeichen versehen werden, sofern sie keine besondere Bedeutung in der Klasse haben; z. B. [\^a], [a\b], [a\]] und [\a].
[^...]	Findet jedes einzelne Zeichen, das <b>nicht</b> in der Klasse vorkommt. [^/]* würde beispielsweise 0 oder mehrere Vorkommen von beliebigen Zeichen finden, das <i>kein</i> Schrägstrich ist, wie z. B. http://. Ebenso würde [^0-9xyz] jedes Zeichen finden, das weder eine Ziffer noch der Buchstabe x, y oder z ist.
\d	Findet jede einzelne Ziffer (entspricht der Klasse [0-9]). Ein großgeschriebenes \D würde hingegen bedeuten, dass jedes Zeichen gefunden wird, das <i>keine</i> Ziffer ist. Diese und die anderen zwei Escapesequenzen darunter können innerhalb einer <u>Klasse</u> verwendet werden; zum Beispiel bedeutet [\d.-]: "jede Ziffer, jeder Punkt oder jedes Minuszeichen".
\s	Findet jedes einzelne Leerraumzeichen wie z. B. Leerzeichen, Tabulatoren und Zeilenumbrüche (`r und `n). Ein großgeschriebenes \S würde hingegen bedeuten, dass jedes Zeichen gefunden wird, das <i>kein</i> Leerraumzeichen ist.
\w	Findet jedes einzelne Wortzeichen, das alphanumerisch oder ein Unterstrich ist. Das ist das gleiche wie [a-zA-Z0-9_]. Ein großgeschriebenes \W würde hingegen bedeuten, dass jedes Zeichen gefunden wird, das <i>kein</i> Wortzeichen ist.
^ \$	Zirkumflex (^) und Dollar-Zeichen (\$) werden <i>Anker</i> genannt. Sie verbrauchen keine Zeichen, sondern binden das Muster an den Anfang oder Ende der zu durchsuchenden Zeichenkette.  ^ kann am Anfang eines Musters stehen, um die Übereinstimmung direkt am Anfang einer Zeile zu erzwingen. Zum Beispiel würde ^abc mit abc123 übereinstimmen, aber nicht mit 123abc.

	<p>\$ kann am Ende eines Musters stehen, um die Übereinstimmung direkt am Ende einer Zeile zu erzwingen. Zum Beispiel würde abc\$ mit 123abc übereinstimmen, aber nicht mit abc123.</p> <p>Die beiden Anker können kombiniert werden. Zum Beispiel würde ^abc\$ nur mit abc übereinstimmen (das heißt, dass sich keine Zeichen davor oder dahinter befinden).</p> <p>Wenn der zu durchsuchende Text mehrere Zeilen enthält, können die Anker mit der <u>m-Option</u> dazu gebracht werden, für jede Zeile zu gelten, anstatt für den gesamten Text. Zum Beispiel würde m)^abc\$ mit 123`r`nabc`r`n789 übereinstimmen. Aber ohne die m-Option würde es nicht übereinstimmen.</p>
<b>\b</b>	<p>\b dient als Abgrenzung eines Wortes. Es ist wie ein Anker, weil es keine Zeichen verbraucht. Es setzt voraus, dass das aktuelle Zeichen, welches ein <u>Wortzeichen</u> (\w) sein muss, das Gegenteil des vorherigen Zeichens ist. Diese Escapesequenz wird normalerweise verwendet, um zu vermeiden, dass das gesuchte Wort innerhalb eines anderen Wortes versehentlich zu finden ist. Zum Beispiel würde \bcat\b nicht mit catfish, sondern mit cat übereinstimmen, egal welche Interpunktionen oder Leerraumzeichen sich daneben befinden. Ein großgeschriebenes \B bewirkt das Gegenteil: Es setzt voraus, dass das aktuelle Zeichen sich <i>nicht</i> neben einer Wortabgrenzung befindet.</p>
<b> </b>	<p>Der senkrechte Strich trennt zwei oder mehr Alternativen. Eine Übereinstimmung erfolgt, wenn <i>eine</i> der Alternativen zutrifft. Zum Beispiel würde gray grey sowohl mit gray als auch mit grey übereinstimmen. Mithilfe der unten beschriebenen Klammern kann das Muster gr(a e)y das gleiche bewirken.</p>