

# Reguläre Ausdrücke (REGEXP)

## Inhaltsverzeichnis

- [1 Übersicht Ausdrücke](#)
- [2 Beispiele](#)
  - [2.1 Erweiterte Liniennummernanzeige](#)
- [3 Links / Quellen](#)

Hier eine Übersicht, welche Zeichen was bedeuten:

## 1 Übersicht Ausdrücke

Zeichen	Funktion / Beschreibung	Beispiel
.	genau <b>ein</b> beliebiges Zeichen für Bereichsangaben	
-	Ist das Minus-Zeichen direkt gemeint und steht nicht am Anfang einer regexp muss [A-Z] ein \ vorangestellt werden: \-	
\	Hebt die (Meta-)Funktion eines Zeichens auf	\. \-
^	Anfang einer Zeichenkette in [ ] negiert es die Aussage: [^a-z] = keine Kleinbuchstaben	^a [^a-z]
\$	Ende einer Zeichenkette	\$a
	Alternativen	[a-z]  [0-9]
+	Zeichen vor dem Plus muss mindestens einmal bis beliebig oft vorkommen	a+
*	Zeichen vor dem Sternchen kann 0 mal bis beliebig oft vorkommen	a*
?	Zeichen vor dem ? muss 0 bis 1 mal vorkommen	a?
{min,max}	Häufigkeit von <i>min</i> bis <i>max</i> : {1,3} = 1-3 mal	\d{1,3}
{anzahl}	bzw. genaue Anzahl: {2} = muss genau 2 mal vorkommen	\d{2}
[ ]	genau ein Zeichen, was zw. den Klammern steht [abc] sucht also nach a, b oder c, nicht nach abc	[A-Z]
( )	Klammern zum Zusammenfassen eines Ausdrucks (wie in der Mathematik): ^(abc) (xyz) = abc oder xyz am Anfang einer Zeichenkette	^(abc) (xyz)
\d	genau eine Ziffer [0-9] (d wie Digit)	\d
\D	ein Zeichen, das keine Ziffer ist, also [^d]	\D
\w	genau ein Buchstabe, eine Ziffer oder der Unterstrich, also [a-zA-Z_0-9] (und evtl. weitere Buchstaben, z. B. Umlaute)	\w
\W	ein Zeichen, das weder Buchstabe noch Zahl noch Unterstrich ist, also [^w] Whitespace; meistens die Klasse der Steuerzeichen \f, \n, \r, \t und \v	\W \s
\S	ein Zeichen, das kein Whitespace ist [^s]	\S

<code>\b</code>	leere Zeichenkette am Wortanfang oder am Wortende (erkennt Anfang eines Wortes)	<code>\b</code>
<code>\B</code>	leere Zeichenkette, die nicht den Anfang oder das Ende eines Wortes bildet	<code>\B</code>
<code>\&lt;</code>	leere Zeichenkette am Wortanfang	<code>\&lt;</code>
<code>\&gt;</code>	leere Zeichenkette am Wortende	<code>\&gt;</code>

## 2 Beispiele

### 2.1 Erweiterte Liniennummernanzeige

Folgender Code erlaubt die einen der folgenden Buchstaben: A E M N S U X vor einer Liniennummer sowie einen der Buchstaben A B C D E hinter der Liniennummer.

Damit sollten die Gängigsten Liniennummern erkannt werden aber andere Kombinationen weggefiltert werden.

Lua

```
filter = "CUSTOM",
params = { expr = "([AEMNSUX]?[0-9]+[ABCDE]?).*$", replace = "\\1", },
```

Erkennt z.B. S1 U3 M17 X54 aber auch 12E

## 3 Links / Quellen

Deutsche Erklärung und Beispiele zu den Bedeutungen:

<https://www.webmaster-seo.com/de/regexp>

Zum Testen der erstellten RegExp:

<https://regex101.com/>

Die wohl nach Urban Games zugrundeliegende Referenz auf Englisch:

<https://www.gnu.org/software/s.../Regular-Expressions.html>