



# GREP-Übersicht

Haeme Ulrich, [www.ulrich-media.ch](http://www.ulrich-media.ch)

Aktualisiert: 10. 06. 2008

Feedback an: [ulrich@ulrich-media.ch](mailto:ulrich@ulrich-media.ch)



# Inhalt

- + Was ist GREP?
- + Wichtige Wildcards
- + Grep Praxisbeispiele

# Was ist GREP?

- + General Regular-Expression Print
- + Suchen/Ersetzen mit «Wildcards», mit «regulären Ausdrücken»
- + Vorteile
  - Nach Mustern suchen
  - Komplexe Suchen/Ersetzen Arbeiten
  - Ersetzt häufig sogar ein Script
- + «Regex» = Abkürzung für «regulärer Ausdruck»

# Wichtige Wildcards

## WIEDERHOLUNG

Wildcard	Beschreibung
?	Null- oder einmal
*	Null- oder mehrmals
+	Ein- oder mehrmals
??	Null- oder einmal. Kürzeste Entsprechung
*?	Null- oder mehrmals. Kürzeste Entsprechung
+?	Ein- oder mehrmals. Kürzeste Entsprechung

## PLATZHALTER

Wildcard	Beschreibung
\d	Beliebige Ziffer (Digit)
.	Beliebiges Zeichen (ohne Umbruchzeichen)
\s	Alle Leerräume
\w	Wortzeichen, keine Leerzeichen (\w+ sucht ein ganzes Wort)
\u	Grossbuchstaben (Uppercase)
\l	Kleinbuchstabe (Lowercase)
~y	Tabulator für Einzug rechts
[[=a=]]	Sucht «a» und alle «a» mit Akzenten: a, ä, â... Statt «a» kann jeder Buchstaben eingegeben werden.

## STELLEN

Wildcard	Beschreibung
<	Wortbeginn
>	Wortende
^	Absatzbeginn
\$	Absatzende
\b	Wortgrenze <code>test\b</code> sucht nach Wörtern, die mit <code>test</code> enden <code>\btest</code> sucht nach Wörtern, die mit <code>test</code> starten
\A	Anfang Textverkettung
\Z	Ende Textverkettung

## ENTSPRECHUNG

Wildcard	Beschreibung
(?<=)	Positive Lookbehind (?<=a)b findet <b>b</b> nur, wenn davor ein <b>a</b> steht
(?<!)	Negatives Lookbehind (?<!a)b findet <b>b</b> nur, wenn davor kein <b>a</b> steht
(?=)	Positives Lookahead a(=?b) findet <b>a</b> nur, wenn danach ein <b>b</b> folgt
(?!)	Negatives Lookahead a(?!b) findet <b>a</b> nur, wenn danach kein <b>b</b> folgt

## KLASSEN

Wildcard	Beschreibung
[a-z]	Sucht alle Buchstaben von a bis z. [0-9] sucht alle Ziffern.

## GEFUNDEN

Wildcard	Beschreibung
\$0	Gesuchtes Element wird durch sich selbst ersetzt
\$1	Erste Gruppe einfügen (\$2 zweite Gruppe, \$3 dritte Gruppe...)

# Grep Praxisbeispiele

Regex	Beschreibung
<code>".+?"</code>	Findet Text in Anführungszeichen. Sucht ein Anführungszeichen, ein beliebiges Zeichen, ein oder mehrmals, kürzeste Distanz zum nächsten Anführungszeichen.
Suchen: <code>^~</code>  Ersetzen: <code>~C</code>	Sucht am Absatzanfang ein Tildezeichen, ersetzt dies mit der Zwischenablage, unformatiert Mit diesem Beispiel können Sie am Absatzbeginn, dort wo ein Tildezeichen gesetzt wurde, zum Beispiel eine Grafik als verankertes Objekt einfügen.
<code>(?&lt;=\\).*(?=\\)</code>	Text in Klammern soll gefunden werden, die Klammern nicht. Sucht beliebigen Text (null oder mehrmals), wenn davor eine öffnende Klammer steht. Sucht die nächst gelegende schliessende Klammer.

Regex	Beschreibung
Suchen: <code>\s+</code>  Ersetzen: <code>\s</code>	Ein- oder mehrere Leerzeichen mit einem ersetzen.
Suchen: <code>(\r)+</code>  Ersetzen: <code>\$1</code>	Ein oder mehrere Returns mit einem ersetzen. Sucht nach einem Return. Mit der Klammer wird das Return gruppiert. Das <code>\$1</code> fügt den Inhalt der Gruppe einmal ein.
Suchen: <code>(&lt;b&gt;)(.+?)(&lt;/b&gt;)</code>  Ersetzen: <code>\$2</code> Zusätzlich: Zeichenformat Ihrer Wahl	HTML-Tag bold mit einem Zeichenformat formatieren. Sucht nach Tag <code>&lt;b&gt;</code> , nach einem beliebigen Zeichen, ein- oder mehrmals, kürzeste Distanz zu <code>&lt;/b&gt;</code> . Erstellt drei Gruppen. Beim Ersetzen wird nur die zweite Gruppe (also ohne Tags) eingefügt.

<b>Regex</b>	<b>Beschreibung</b>
<p data-bbox="84 140 180 210">Suchen: (.\Z)</p> <p data-bbox="84 255 193 325">Ersetzen: \$1~c</p>	<p data-bbox="544 140 1476 244">Ein Sonderzeichen am Ende der Textverkettung einfügen.</p> <p data-bbox="544 249 1505 420">Sucht irgend ein Zeichen, gefolgt vom Ende der Textverkettung. Setzt dieses letzte Zeichen der Textverkettung in eine Gruppe (die Klammer). Eingefügt wird die Gruppe (das letzte Zeichen), gefolgt vom formatierten Inhalt der Zwischenablage.</p>

Regex	Beschreibung
<p>Suchen: (\)(.+?)(\)</p> <p>Ersetzen: \$2</p> <p>Zusätzlich: Zeichenformat Ihrer Wahl</p>	<p>Text in eckiger Klammer soll ein Zeichenformat zugewiesen werden. Zudem sollen die eckigen Klammern entfernt werden.</p> <p>Beispiel vorher: Das ist eine Demo [Objekt &gt; Textrahmenoptionen] und da kommt wieder Text.</p> <p>Beispiel nachher: Das ist eine Demo <b>Objekt &gt; Textrahmenoptionen</b> und da kommt wieder Text.</p> <p>Erklärung: Die drei Klammern machen drei Gruppen. Eingefügt wird nur die zweite Gruppe, die eckigen Klammern verschwinden also beim Einfügen.</p> <p>Gesucht wird nach eckiger Klammer. Der Backslash vor der Klammer muss sein, damit die Klammer als Zeichen und nicht als Funktion verstanden wird. Dann wird mit dem Punkt nach irgend einem Zeichen gesucht, das ein- oder mehrmals vorkommt. Dann durch das Fragezeichen die kürzeste Distanz zur schliessenden Klammer.</p>

Regex	Beschreibung
Suchen: <code>\s1 d1 1</code>  Ersetzen: Format ersetzen > Laufweite	Alle 1 sollen unterschritten werden: 121, 212, 111, 123, 321, 143, 1999, 2001, 2010, 1 Sucht: 1 Leerzeichen gefolgt von einer 1, oder eine Zahl gefolgt von einer 1, oder eine 1 Ersetzen mit «Format ersetzen:» oder ein Zeichenformat zuweisen.
Suchen: <code>\u\u+</code>	Sucht alle versal geschriebenen Wörter. \u = uppercase (gross geschrieben). Das zweite \u mit dem + sagt, dass ein oder mehrere weitere Grossbuchstaben folgen.