

# Reguläre Ausdrücke in der Arbeit mit Textdaten

Reguläre Ausdrücke werden verwendet, wenn man eine ungefähre Vorstellung davon hat, was es in einer Datei zu finden gibt, ohne genau zu wissen, wie das Gesuchte im Detail aussieht. Doch wie hat man sich so etwas vorzustellen? Wie lassen sich Zeichen und Zeichenfolgen umschreiben? Welche Kriterien können herangezogen werden, um Zeichen und abstrakte Muster zu beschreiben? Und wie lassen sich sich aus konkreten Textdaten überhaupt Muster abstrahieren und mithilfe regulärer Ausdrücke formalisieren?

Zu abstrakt? Fangen wir klein an: Das wohl hilfreichste Metazeichen (= Zeichen, die nicht für sich selbst stehen, sondern bestimmte Funktionen erfüllen/ Prozesse beschreiben) ist der Punkt (".") und dient als Platzhalter für ein beliebiges Zeichen. Um also nicht nach *ragazza/ragazze* einzeln suchen zu müssen, kann mithilfe der regulären Ausdrücke alternativ der Suchausdruck *ragazz.* verwendet werden. Weitere reguläre Ausdrücke sind Positionsmarker, wie "^", welches markiert, dass das entsprechende Wort am Zeilenanfang stehen muss (z.B. *^ragazza*), und Wiederholungsoperatoren, z.B. "\*", der in etwa aussagt: "Wiederhole das vorausgehende (Meta-)Zeichen beliebig oft" (wie auch in: *Wo\*w*, ich versteh's = "Wow, ich versteh's", "Woow, ich versteh's", "Woowooooooooow, ich versteh's").

Alles, was man nach diesen einleitenden Informationen für den Start braucht, sind ein Regex-fähiger Texteditor und das vorliegende Manual. Texteditoren existieren viele, zum Beispiel TextWrangler (MAC), jEdit oder das häufig vorinstallierte Notepad++. Wir empfehlen Textpad in der neusten Version (aktuell 7.5.1) (in den älteren Versionen funktionieren einige reguläre Ausdrücke nicht, andere werden komplizierter ausgedrückt). Textpad gibt es als Testversion gratis.

Und ein wichtiger Hinweis am Schluss: Am besten nicht verzweifeln!! Unterschiedliche Texteditoren funktionieren nicht immer mit denselben regulären Ausdrücken, teilweise nicht einmal die verschiedenen Versionen desselben Editors. Aus diesem Grund präsentieren wir, wo möglich, Alternativen zu nicht universell gebrauchten Ausdrücken. Und auch das wird das Problem nicht immer lösen. Im Worst Case Scenario können ein paar Backslashes mehr oder weniger helfen, aber mehr dazu später. Mit der Erfahrung wächst die Fähigkeit, Fehler zu lokalisieren und sie einem anderen zugrundeliegenden Standard anzulasten. Deshalb: Am besten nicht verzweifeln...

- [Vorab](#)
- [Reg-was?](#)
- [Von null bis unendlich](#)
- [Platzhalter reloaded](#)
- [Escape](#)
- [Und oder](#)
- [Ersatz gefunden](#)

[Powerpoint-Präsentation zur Einführung](#)