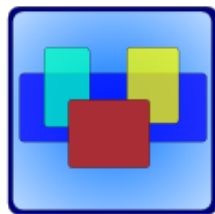


# Das Handbuch zum Editor für reguläre Ausdrücke

Jesper K. Pedersen  
Übersetzung: Gregor Zumstein



## Das Handbuch zum Editor für reguläre Ausdrücke

# Inhaltsverzeichnis

<b>1</b>	<b>Einführung</b>	<b>5</b>
<b>2</b>	<b>Was ein regulärer Ausdruck ist</b>	<b>6</b>
<b>3</b>	<b>Benutzung des Editors für reguläre Ausdrücke</b>	<b>9</b>
3.1	Aufbau der Benutzungsschnittstelle . . . . .	9
3.2	Bearbeitungswerkzeuge . . . . .	10
3.2.1	Auswahlwerkzeug . . . . .	10
3.2.2	Textwerkzeug . . . . .	10
3.2.3	Buchstabenwerkzeug . . . . .	10
3.2.4	Werkzeug für beliebige Zeichen . . . . .	11
3.2.5	Wiederholungswerkzeug . . . . .	11
3.2.6	Alternativen-Werkzeug . . . . .	11
3.2.7	Zusammensetzungswerkzeug . . . . .	12
3.2.8	Zeilenanfang/-ende-Werkzeuge . . . . .	12
3.2.9	(Nicht-)Wortgrenzen-Werkzeuge . . . . .	12
3.2.10	Positiv/Negativ vorausschauende Werkzeuge . . . . .	13
3.3	Benutzerdefinierte reguläre Ausdrücke . . . . .	13
<b>4</b>	<b>Berichten von Fehlern und Wünschen</b>	<b>14</b>
<b>5</b>	<b>Häufig gestellte Fragen</b>	<b>15</b>
5.1	Unterstützt der Editor für reguläre Ausdrücke Rückwärtsreferenzen? . . . . .	15
5.2	Unterstützt der Editor für reguläre Ausdrücke die Anzeige von Übereinstimmungen? . . . . .	15
5.3	Ich bin Entwickler einer KDE-Anwendung. Wie kann ich dieses GUI-Element in meiner Anwendung benutzen? . . . . .	15
5.4	Ich kann den Knopf <i>Regulären Ausdruck bearbeiten</i> in Konqueror oder einer anderen KDE-Anwendung nicht finden. . . . .	15
<b>6</b>	<b>Danksagungen und Lizenz</b>	<b>16</b>

## **Zusammenfassung**

Dieses Handbuch beschreibt die Benutzung des Editors für reguläre Ausdrücke

# Kapitel 1

## Einführung

Mit dem Editor für reguläre Ausdrücke können reguläre Ausdrücke mit grafischer Unterstützung zusammengestellt werden und müssen nicht mit komplizierten und kryptischen Zeichenfolgen definiert werden (Dies sieht zum Beispiel etwa so aus: `^.*kde\b.`.) Die hauptsächlichsten Nachteile dieser Art sind:

- Es ist für Nicht-Programmierer schwierig zu verstehen.
- Sie müssen manche Zeichen speziell kennzeichnen (um zum Beispiel einen Stern zu finden, müssen Sie `\*` tippen).
- Sie müssen sich die Regeln für die *Reihenfolge* merken. (Was trifft auf `x|y*` zu? Ein einzelnes `x` oder eine Anzahl `y`, ODER eine Anzahl `x` und `y`'s gemischt?)

Mit dem Editor für reguläre Ausdrücke können Sie reguläre Ausdrücke auf eindeutige Weise *zeichnen*. Der Editor löst zumindest die beiden letzten Probleme. Ob dieser Editor reguläre Ausdrücke für Nicht-Programmierer zugänglich macht, weiß der Autor nicht. Falls Sie aber ein Nicht-Programmierer sind, der dank diesem Editor reguläre Ausdrücke verwenden kann, schicken sie [ihm](#) bitte eine kurze Nachricht.

## Kapitel 2

# Was ein regulärer Ausdruck ist

Ein regulärer Ausdruck ist ein Weg *Bedingungen* anzugeben, die für eine bestimmte Situation erfüllt sein müssen. Im normalen Suchdialog eines Texteditors können sie nur nach *genau* der eingegebenen Zeichenfolge suchen. Mit einem regulären Ausdruck hingegen geben Sie an wie das Suchergebnis aussehen soll. Beispiele hierfür sind: Suche das Wort *KDE*, aber nur am Zeilenanfang. Oder suche das Wort *der*, aber nur wenn es allein steht. Oder suche alle Dateien, die mit *test* beginnen und danach eine beliebige Anzahl Ziffern haben, wie *test107* und *test007*.

Reguläre Ausdrücke werden zusammengesetzt aus kleineren regulären Ausdrücken, genau so wie Lego-Bauwerke aus kleineren Teilen zusammengesetzt werden. Genau wie bei Lego gibt es auch für reguläre Ausdrücke elementare Bausteine. Diese werden im folgenden anhand von Beispielen vorgestellt.

---

### Example 2.1 Suche nach normalem Text.

Wenn nur nach einem bestimmten Text gesucht werden soll, sind reguläre Ausdrücke keine sonderlich gute Wahl. Grund dafür ist, dass in regulären Ausdrücken gewisse Zeichen eine Spezialbedeutung besitzen. Diese sind `. * | $`. Wenn also nach dem Text `kde`. (also den Zeichen `kde` gefolgt von einem Punkt), gesucht werden soll, muss dies als `kde\.` angegeben werden.<sup>1</sup> Das Schreiben von `\.` anstelle von `.` nennt man *Escaping* oder *Maskieren*.

---

---

### Example 2.2 Finden von URLs

Wenn Sie in KDE etwas das wie eine URL aussieht auswählen, so ermöglicht Ihnen Klipper, diese ausgewählte URL mit Konqueror zu öffnen.

Klipper ermöglicht dies, indem es den ausgewählten Text mit mehreren verschiedenen regulären Ausdrücken vergleicht und bei Übereinstimmung eine der URL entsprechende Funktion anbietet.

Der reguläre Ausdruck für URLs sagt (unter anderem), dass die Adresse mit `http://` starten muss. Dies wird durch den regulären Ausdruck erreicht, der ein Dach `^` vor das `http://` setzt.

Die erwähnte Anwendung ist ein Beispiel bei dem die Position übereinstimmen muss. Auf ähnliche Weise kann auch das *Ende der Zeile* durch das `$`-Zeichen gefunden werden.

---

---

**Example 2.3** Suche nach dem Wort *der*, aber nicht *oder*, *gliedert* oder *dereinst*.

Zwei weitere Grenzen können auf gleiche Weise gefunden werden. Einerseits die *Wortgrenze* andererseits die *Nicht-Wort-Grenze*. Diese werden durch `\b` für die Wortgrenze und `\B` für die Nicht-Wort-Grenze gekennzeichnet.

So kann also die Suche nach dem Wort *der* durch den regulären Ausdruck `\bthe\b` erfolgen. Dies bedeutet, dass wir nach dem Wort *der* ohne Buchstaben hinten und vorne (also einer Wortgrenze) suchen.

Die vier Grenzen für reguläre Ausdrücke werden im Editor für reguläre Ausdrücke über die vier [Grenzen-Werkzeuge](#) eingefügt.

---

---

**Example 2.4** Suche nach entweder *dies* oder *das*

Angenommen Sie wollen nach den beiden Wörtern *dies* und *das* suchen. Mit einer normalen Suchfunktion sind Sie gezwungen, dies in zwei Schritten durchzuführen: Im ersten suchen Sie nach *dies*, im zweiten nach *das*.

Mit der Benutzung von regulären Ausdrücken, ist es möglich, gleichzeitig nach beiden Begriffen zu suchen. Hierzu wird der Ausdruck `dies|das` verwendet, die beiden gesuchten Wörter also durch einen vertikalen Strich getrennt.<sup>2</sup>

Im Editor für reguläre Ausdrücke schreiben Sie den senkrechten Strich nicht selbst, sondern wählen stattdessen das [Alternativen-Werkzeug](#) und fügen die verschiedenen Teile (die selbst reguläre Ausdrücke sind) übereinander.

---

---

**Example 2.5** Finden von irgendwas

Reguläre Ausdrücke werden oft mit den Platzhaltern auf der Befehlszeile verglichen, also der Möglichkeit eine Menge von Dateien mit dem Stern (\*) oder dem Fragezeichen (?) einzuschränken. Sie kennen sicher die Platzhalter in den folgenden Beispielen:

- `ls *.txt` - Hier ist `*.txt` der Platzhalter für alle Dateien, die mit der Erweiterung `.txt` enden.
- `cat test?? .res` - wird auf alle Dateien angewandt, die mit `test` beginnen, dann zwei beliebige Zeichen enthalten, auf die die Dateierweiterung `.res` folgt.

Auf der Befehlszeile entspricht der Stern (\*) einer beliebigen Anzahl beliebiger Zeichen. Mit anderen Worten, der Stern steht für *Alles*. Bei regulären Ausdrücken erreichen sie diesen Effekt mit der Zeichenfolge `.*`. Der Punkt entspricht einem beliebigen Zeichen, also jeweils nur einem *einzigem* Zeichen und der Stern besagt, dass der reguläre Ausdruck davor beliebig oft wiederholt werden kann. Zusammen ergibt das also eine beliebige Anzahl beliebiger Zeichen.

Auf den ersten Blick erscheint dies übermäßig kompliziert, eröffnet aber nahezu unbegrenzte Möglichkeiten. Ein weiterer einfacher regulärer Ausdruck ist `a`. Der Buchstabe `a` stellt einen regulären Ausdruck dar, der einem einzelnen Buchstaben entspricht, dem Buchstaben `a`. Wenn wir diesen mit einem Stern (\*) kombinieren, also `a*`, haben wir einen regulären Ausdruck, der jeder beliebigen Anzahl 'a's entspricht.

Es ist möglich, mehrere reguläre Ausdrücke hintereinander zu stellen, z.B. `ba(na)*`.<sup>3</sup> Mit diesem Ausdruck würden Sie also unter anderem die folgenden Texte finden: `ba`, `bana`, `banana`, `bananan`, `ananananana`.

Mit diesen Informationen ist es jetzt nicht schwer, die Befehlszeilen-Platzhalter `test?? .res` als regulären Ausdruck nachzubilden. Dieser heißt `test.\.res`. Der einzelne Punkt ist jedes beliebige Zeichen. Um einen einzelnen Punkt zu finden, muss `\.` angegeben werden.<sup>4</sup> Mit anderen Worten findet der reguläre Ausdruck `\.` einen Punkt, während der einzelne Punkt jedes Zeichen findet.

Im Editor für reguläre Ausdrücke wird ein wiederholter regulärer Ausdruck mit dem [Wiederholungs-Werkzeug](#) erstellt.

---

---

**Example 2.6** Ersetzen von & mit &amp; in einem HTML-Dokument

---

In HTML muss das Zeichen & als &amp; geschrieben werden. Das ist ähnlich wie das Escapen in einem regulären Ausdruck.

Stellen Sie sich vor, Sie hätten in einem gewöhnlichen Editor (z.B. Kate oder XEmacs) ein HTML-Dokument erstellt und hätten dabei diese Regel vollkommen vergessen. In diesem Fall müssten Sie also alle Vorkommen von & durch &amp; ersetzen.

Dies können Sie einfach mit einem gewöhnlichen Ersetzen erledigen. Es gibt aber ein kleines Problem. Wenn Sie manchmal doch an die Regel gedacht haben, funktioniert das nicht. Ein gewöhnliches Ersetzen würde zur Folge haben, dass &amp; durch &amp;amp; ersetzt würde.

Sie wollen aber eigentlich, dass alle Vorkommen von & ersetzt werden, die *nicht* von den Buchstaben amp; gefolgt sind. Dies wird mit *vorausschauenden* regulären Ausdrücken erreicht.

Der reguläre Ausdruck, der ein kaufmännisches Und nur dann findet, wenn es nicht von den Buchstaben amp; gefolgt ist, sieht wie folgt aus: & (?!amp;) Es ist natürlich leichter, diesen Ausdruck im Editor zu lesen, wo dafür die [vorausschauenden Werkzeuge](#) zur Verfügung stehen.

---

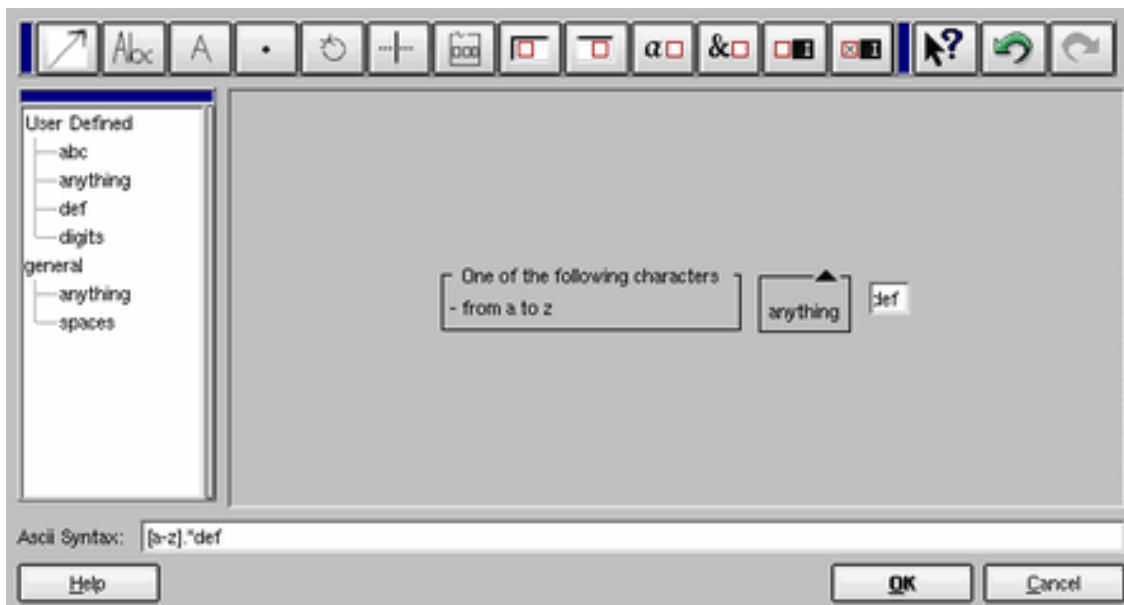


## Kapitel 3

# Benutzung des Editors für reguläre Ausdrücke

Dieses Kapitel erklärt die Arbeitsweise des Editors für reguläre Ausdrücke.

### 3.1 Aufbau der Benutzungsschnittstelle



Der bedeutendste Teil des Editors ist natürlich der Arbeitsbereich, wo ein regulärer Ausdruck gezeichnet wird.

Oberhalb des Arbeitsbereichs gibt es zwei Werkzeugleisten, die erste enthält [Bearbeitungsaktionen](#) - ähnlich den Werkzeugen in einem Zeichenprogramm. Die zweite Werkzeugleiste enthält den Knopf für die *Was ist das?-Hilfe* und Knöpfe für Rückgängig und Wiederherstellen.

Unterhalb des Arbeitsbereichs steht der aktuell bearbeitete reguläre Ausdruck in seiner ASCII-Notation. Diese wird laufend nachgeführt, wenn im grafischen Editor eine Änderung vorgenommen wird. Umgekehrt können auch Änderungen in der ASCII-Notation vorgenommen werden, die dann ebenfalls unmittelbar auch im grafischen Editor dargestellt werden.

Schlussendlich finden sich auf der linken Seite des Eingabebereichs einige vordefinierte reguläre Ausdrücke. Dafür gibt es zwei Gründe: (1) Wird ein vordefinierter regulärer Ausdruck verwendet wird er verständlicher dargestellt. So wird zum Beispiel der reguläre Ausdruck „\*“ durch ein Symbol mit dem Text „anything“ dargestellt. (2) Wenn Sie einen regulären Ausdruck erzeugen, können Sie möglicherweise aus der Menge der vordefinierten Ausdrücke einige direkt benutzen. Wie Sie selbst definierte reguläre Ausdrücke speichern können, sehen sie unter [Benutzerdefinierte reguläre Ausdrücke](#).

## 3.2 Bearbeitungswerkzeuge

Für diesen Abschnitt ist es erforderlich, dass Sie das Kapitel [Was ist ein regulärer Ausdruck](#) schon gelesen haben oder über äquivalentes Wissen verfügen.

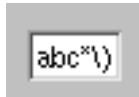
Alle Bearbeitungswerkzeuge befinden sich oberhalb des Arbeitsbereichs. Im folgenden wird jedes einzelne davon beschrieben.

### 3.2.1 Auswahlwerkzeug



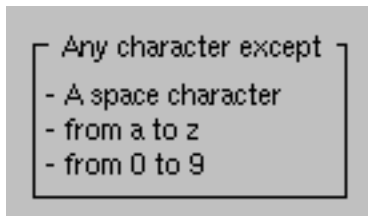
Das Auswahlwerkzeug wird gebraucht, um Elemente auszuwählen, welche ausgeschnitten oder verschoben werden sollen. Es funktioniert gleich wie in Zeichenprogrammen.

### 3.2.2 Textwerkzeug



Dieses Werkzeug dient dem Einfügen von gewöhnlichem Text. Der Text wird Zeichen für Zeichen auf Gleichheit überprüft, es müssen hierfür auch keine Zeichen speziell gekennzeichnet werden. Im Beispiel oben wird der folgende reguläre Ausdruck erstellt: `abc*\\)`

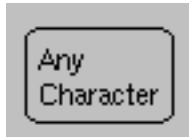
### 3.2.3 Buchstabenwerkzeug



Mit diesem Werkzeug werden Zeichenbereiche eingefügt. Zum Beispiel alle Zeichen zwischen „a“ und „z“ oder „0“ und „9“. Bei der Auswahl dieses Werkzeugs erscheint ein Dialog, in dem der Zeichenbereich eingegeben werden kann.

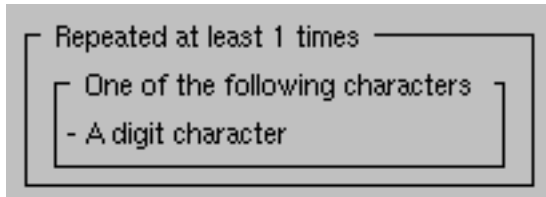
Siehe die Beschreibung von [wiederholten regulären Ausdrücken](#).

### 3.2.4 Werkzeug für beliebige Zeichen



Dies ist der reguläre Ausdruck „Punkt“ (.). Er entspricht jedem beliebigen Zeichen.

### 3.2.5 Wiederholungswerkzeug



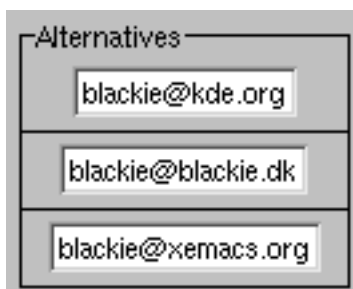
Dieses Werkzeug dient dem Wiederholen von Teilausdrücken. Darunter sind die regulären Ausdrücke „Stern“ (\*), „Plus“ (+), „Fragezeichen“ (?) und Bereiche ({3,5}). Beim Einfügen mit diesem Werkzeug erscheint ein Dialogfenster, in dem die Anzahl Wiederholungen eingegeben werden kann.

Was wiederholt werden soll, kommt im Innern dieses Elements zu stehen.

Wiederholte Teilausdrücke können sowohl von innen nach außen als auch von außen nach innen erzeugt werden. Es spielt also keine Rolle, ob zuerst die Wiederholung oder der zu wiederholende Teilausdruck eingefügt wird.

Siehe die Beschreibung von [Wiederholten regulären Ausdrücken](#).

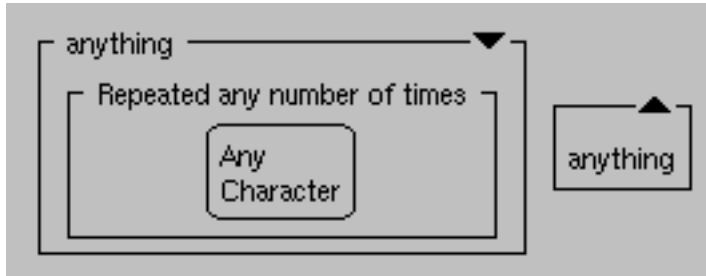
### 3.2.6 Alternativen-Werkzeug



Dies ist der reguläre Ausdruck der Alternative (|). Alle Alternativen werden übereinander gezeichnet.

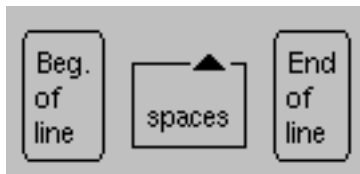
Siehe die Beschreibung des [Alternativen regulären Ausdrucks](#)

### 3.2.7 Zusammensetzungswerkzeug



Das Zusammensetzungswerkzeug stellt keinen regulären Ausdruck dar. Es wird stattdessen gebraucht, um Teilausdrücke zusammenzufassen und diese nur durch einen Titel gekennzeichnet, darzustellen.

### 3.2.8 Zeilenanfang/-ende-Werkzeuge



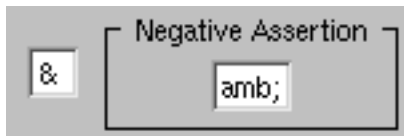
Die Zeilenanfangs und -ende-Werkzeuge finden den Beginn respektive das Ende einer Zeile. Siehe die Beschreibung von [Ortsbezogenen regulären Ausdrücken](#).

### 3.2.9 (Nicht-)Wortgrenzen-Werkzeuge



Die Wortgrenzen-Werkzeuge finden Wortgrenzen respektive Nicht-Wortgrenzen. Siehe die Beschreibung von [Wortgrenzen in regulären Ausdrücken](#)

### 3.2.10 Positiv/Negativ vorausschauende Werkzeuge



Die vorausschauenden Werkzeuge definieren entweder einen positiven oder negativen regulären Ausdruck. Der gefundene Teilausdruck selbst ist jedoch nicht Teil der gesamten gefundenen Zeichenfolge.

Bemerkung: Vorausschauende reguläre Ausdrücke dürfen nur am Ende von regulären Ausdrücken verwendet werden. Der Editor für reguläre Ausdrücke erzwingt dies jedoch nicht.

Siehe die Beschreibung von [vorausschauenden regulären Ausdrücken](#).

## 3.3 Benutzerdefinierte reguläre Ausdrücke

Links vom Arbeitsbereich ist eine Liste mit benutzerdefinierten regulären Ausdrücken. Einige vordefinierte reguläre Ausdrücke werden mit KDE mitgeliefert, weitere können Sie selbst speichern.

Mit diesen regulären Ausdrücken wird zweierlei erreicht ([Details](#)). Erstens werden Bausteine für eigene reguläre Ausdrücke zur Verfügung gestellt und zweitens werden häufig gebrauchte reguläre Ausdrücke schöner dargestellt.

Ihren eigenen regulären Ausdruck speichern Sie, indem Sie im Arbeitsbereich auf die rechte Maustaste klicken und `Regulären Ausdruck speichern` wählen.

Wenn der zu speichernde reguläre Ausdruck eine [Zusammensetzung](#) ist, so werden reguläre Ausdrücke, die ihn verwenden, ebenfalls schöner dargestellt.

Benutzerdefinierte reguläre Ausdrücke können gelöscht oder umbenannt werden, indem mit der rechten Maustaste das Kontextmenü geöffnet und der entsprechende Menüeintrag ausgewählt wird.

## Kapitel 4

# Berichten von Fehlern und Wünschen

Fehlermeldungen und Wünsche zur Funktionalität werden mit der [KDE-Fehlerverfolgung](#) gemeldet. **Bevor** Sie eine Fehlermeldung abschicken oder eine Funktionalitätserweiterung wünschen, prüfen Sie bitte, ob nicht schon jemand dasselbe [bemerkt](#) oder [gewünscht](#) hat.

## Kapitel 5

# Häufig gestellte Fragen

### 5.1 Unterstützt der Editor für reguläre Ausdrücke Rückwärtsreferenzen?

Im Moment wird dies nicht unterstützt. Es ist für die nächste Version geplant.

### 5.2 Unterstützt der Editor für reguläre Ausdrücke die Anzeige von Übereinstimmungen?

Nein, es ist Ziel, dies in der nächsten Version zur Verfügung zu stellen.

### 5.3 Ich bin Entwickler einer KDE-Anwendung. Wie kann ich dieses GUI-Element in meiner Anwendung benutzen?

Vergleichen Sie die Dokumentation für die Klasse [KRegExpEditorInterface](#).

### 5.4 Ich kann den Knopf *Regulären Ausdruck bearbeiten* in Konqueror oder einer anderen KDE-Anwendung nicht finden.

Der Editor für reguläre Ausdrücke befindet sich im Paket `kdeutils`. Ist dieses nicht installiert, gibt es auch den Knopf *Regulären Ausdruck bearbeiten* nicht.

## Kapitel 6

# Danksagungen und Lizenz

Copyright für die Dokumentation 2001, Jesper K. Pedersen [blackie@kde.org](mailto:blackie@kde.org)

Übersetzung Gregor Zumstein [zumstein@ssd.ethz.ch](mailto:zumstein@ssd.ethz.ch)

Diese Dokumentation ist unter den Bedingungen der [GNU Free Documentation License](#) veröffentlicht.

Dieses Programm ist unter den Bedingungen der [GNU General Public License](#) veröffentlicht.