Table of Contents

Regular Expressions

Introduction

Regular expressions (REs), unlike simple queries, allow you to search for text which matches a particular pattern.

REs are similar to (but more poweful than) the "wildcards" used in the command-line interfaces found in operating systems such as Unix and MS-DOS. REs are used by sophisticated search engines, as well as by many Unix-based languages and tools (e.g., `awk`, `grep`, `lex`, `perl`, and `sed`).

Examples

`compan(y|ies)`	Search for **company**, **companies**
`(peter|paul)`	Search for **peter**, **paul**
`bug*`	Search for **bug**, **bugg**, **buggg** or simply **bu** (a star matches **zero** or more instances of the previous character)
`bug.*`	Search for **bug**, **bugs**, **bugfix** (a dot-star matches zero or more instances of **any** character)
`[Bb]ag`	Search for **Bag**, **bag**
`b[aiueo]g`	Second letter is a vowel. Matches **bag**, **bug**, **big**
`b.g`	Second letter is any letter. Matches also **b&g**
`[a-zA-Z]`	Matches any one letter (but not a number or a symbol)
`[^0-9a-zA-Z]`	Matches any symbol (but not a number or a letter)
`[A-Z][A-Z]*`	Matches one or more uppercase letters
`[0-9]{3}-[0-9]{2}-[0-9]{4}`	US social security number, e.g. **123-45-6789**
`Chart;PNG;!JPG`	Search for topics containing the words **Chart** *and* **PNG** *and not* **JPG**. The `";"` and `"!"` are TWiki-specific operators and are not part of regular expression

Searches with "and" combinations

A TWiki `type="regex"` SEARCH introduces two special characters to regular expressions:

- The semicolon `";"` is an AND-operator.
  Example search for "form" *and* "template": `form;template`
- The exclamation point `"!"` as a NOT-operator.
  Example search for "soap" *and* "wsdl" *and not* "shampoo": `soap;wsdl;!shampoo`

TWiki also supports a SQL-like query search where you can literally use AND and NOT operators.

Use Google if your TWiki site is public. Example search for "form" *and* "template" at TWiki.org:
`site:twiki.org +form +template`

Advanced

Here is stuff for our UNIX freaks: (copied from 'man egrep')

> A regular expression is a pattern that describes a set of strings. Regular expressions are constructed analogously to arithmetic expressions, by using various operators to combine

smaller expressions.

The fundamental building blocks are the regular expressions that match a single character. Most characters, including all letters and digits, are regular expressions that match themselves. Any metacharacter with special meaning may be quoted by preceding it with a backslash.

A bracket expression is a list of characters enclosed by [and]. It matches any single character in that list; if the first character of the list is the caret ^ then it matches any character not in the list. For example, the regular expression [0123456789] matches any single digit.

Within a bracket expression, a range expression consists of two characters separated by a hyphen. It matches any single character that sorts between the two characters, inclusive, using the locale's collating sequence and character set. For example, in the default C locale, [a-d] is equivalent to [abcd]. Many locales sort characters in dictionary order, and in these locales [a-d] is typically not equivalent to [abcd]; it might be equivalent to [aBbCcDd], for example.

Finally, certain named classes of characters are predefined within bracket expressions, as follows. Their names are self explanatory, and they are [:alnum:], [:alpha:], [:cntrl:], [:digit:], [:graph:], [:lower:], [:print:], [:punct:], [:space:], [:upper:], and [:xdigit:]. For example, [[:alnum:]] means [0-9A-Za-z], except the latter form depends upon the C locale and the ASCII character encoding, whereas the former is independent of locale and character set. (Note that the brackets in these class names are part of the symbolic names, and must be included in addition to the brackets delimiting the bracket list.) Most metacharacters lose their special meaning inside lists. To include a literal] place it first in the list. Similarly, to include a literal ^ place it anywhere but first. Finally, to include a literal - place it last.

The period . matches any single character. The symbol \w is a synonym for [[:alnum:]] and \W is a synonym for [^[:alnum]].

The caret ^ and the dollar sign $ are metacharacters that respectively match the empty string at the beginning and end of a line. The symbols \< and \> respectively match the empty string at the beginning and end of a word. The symbol \b matches the empty string at the edge of a word, and \B matches the empty string provided it's not at the edge of a word.

A regular expression may be followed by one of several repetition operators:

?	The preceding item is optional and matched at most once.
*	The preceding item will be matched zero or more times.
+	The preceding item will be matched one or more times.
{n}	The preceding item is matched exactly n times.
{n,}	The preceding item is matched n or more times.
{n,m}	The preceding item is matched at least n times, but not more than m times.

Two regular expressions may be concatenated; the resulting regular expression matches any string formed by concatenating two substrings that respectively match the concatenated subexpressions.

Two regular expressions may be joined by the infix operator |; the resulting regular expression matches any string matching either subexpression.

Repetition takes precedence over concatenation, which in turn takes precedence over alternation. A whole subexpression may be enclosed in parentheses to override these precedence rules.

The backreference \n, where n is a single digit, matches the substring previously matched by the nth parenthesized subexpression of the regular expression.

Related Links:

- http://perldoc.perl.org/perlretut.html - Regular expressions tutorial
- http://www.perl.com/doc/manual/html/pod/perlre.html - Perl regular expressions

Related Topics: UserDocumentationCategory, VarSEARCH, FormattedSearch, QuerySearch

-- ***Contributors:*** PeterThoeny, JohnTalintyre

This topic: TWiki21Nov > RegularExpression
Topic revision: r9 - 2010-05-04 - TWikiContributor