Quickstart – Regular Expressions

A regular expression is a text pattern consisting of ordinary characters and **meta characters/ symbols** which is matched against simple strings. In order to find complex patterns in the corpus, you can use regular expressions as search expressions. The purpose is optimizing symbol based operations (especially search operations), i.e. finding strings in larger strings or texts without listing all alternatives separately. Examples:

- The words *this, that, those* and *these* and their capitalized variants:
  [Tt]h(is|at|ose|ese)
- Words starting with *in* and ending in *able* or *ably* (e.g. *indisputable, indescribably, ineffable, indistinguishable* etc.):
  \bin[a-z]+abl[ey]\b
- All sequences of three words followed by a question mark, i.e. the last three words of questions
  (\b[A-Za-z]+\b){3,3}\?

A. Symbols and symbol classes

a	matches the **symbol** a (case sensitive!)
.	matches **any symbol**
[ui]	**symbol class**, matches “u” or “i”
[^AaEeIiOoUu]	**negated symbol class** matches all symbols except vowels
[A-Ka-k] [0-5]	**range** of symbols

Examples:
B[ui]rma → matches “Burma” and “Birma”
M[ae][iy]er → matches “Maier”, “Mayer”, “Meyer”, “Meier”

B. Predefined symbol classes (selection)

\d	a digit
\D	a non-digit
\w	an alphabetic symbol
\W	a non-alphabetic symbol
\s	whitespace
\S	non-whitespace

Important!!!
\w → matches letters of the **English** alphabet, no extensions of the Latin alphabet
Instead: **[A-Za-zÄäÖöÜüß]** for the German alphabet

C. Quantifiers

?	once or not at all
*****	zero or more times
+	once or more times
{n}	*n* times
{min,max}	at least *min* times, at most *max* times

Examples:
[A-K]\w+ → matches words with capital initial A-K
\d{1,2}\.\d{1,2}\.\d{2,4} → matches dates (of format: 14.11.1971)

D. Other metasymbols (selection)

|	alternative (OR)
(…|…)	Groups expressions
\b	matches a word boundary
\	escape (literal interpretation)

Examples:
\. → matches a period
[Dd](er|ie|as) → matches German definite articles
[a-z0-9-_\.]+@[a-z0-9-_\.]+\.\w{2,3} → matches eMail addresses

By combining regular expressions in various ways, you can formulate rather complex queries with them. Useful as they are, regular expressions are not very easy to learn. There are very many books and websites explaining regular expressions. We recommend that you consult at least one of those and use it as a reference when working with EXAKT.

For those not afraid of formal specifications, the exact syntax and usage of regular expressions is explained at:

http://java.sun.com/javase/6/docs/api/java/util/regex/Pattern.html