Regular Expressions

A regular expression is a description of a pattern that describes a set of possible characters in an input string.

Upside

Useful.

Downside

Not consistent from implementation to implementation.

Syntax

The simplest is a single character. `c` will match one occurance of a c. A `.` will stand for any character (with some exceptions that I've already forgotten).

`^` specifies the beginning of a line. `^The` will match any 'The' that's at the beginning of a line.

`$` specifies the end of a line. `well$` will match any 'well' that's at the end of a line.

`^Bin$` specifies a line that only contains 'Bin'.

Character Classes

Square brackets are used to define character classes.

`[aeiou]` will match any of the characters 'a', 'e', 'i', 'o', 'u'.

Ranges

`[0-9]` matches 0, 1, 2, 3, 4, 5, 6, 7, 8, or 9. You can't use a range backwards. If you put in e.g. `[e-a]`, this will match e, -, or a.

Negation

Putting a `^` at the start of a character class will negate it. `[^aeiou]` will match any letter except a, e, i, o, or u.

Alternation

`(T|P)` will match T or P. `At(ten|nine)tion` will match 'Attention' or 'Atninetion'.

Quantifiers

Optional Items

A `?` makes the character or bracketed expression preceeding it optional, it will match things where it's present and things where it's not.

Repetition

An `*` will match 0 or more of the preceeding character or bracketed expression.

A `+` will match 1 or more of the preceeding character of bracketed expression.

Repetition Ranges

Look it up.

Backreferences

Look it up.