

Core regular expressions reference sheet

Quantifiers

- + one or more of the preceding
- ? one or none of the preceding
- * zero or more of the preceding

Anchors

- ^ beginning of the line
- \$ end of the line

Escaping

The following special characters need \ to make them literal

`$^.[]+?*\\`

Sometimes also: & () {} |

Character classes

Anything within [] means 'any one of these', eg

`[cat]` = 'c' or 'a' or 't'

`[a-z]` = any letter

`[0-9]` = any number

Negate with an initial ^, eg `[^;]` = 'anything not a semi-colon'

Special characters

`.` any character

`\n` line break

`\t` tab

Back referencing in replacements

This is the main area of variety between software

Either () or \(\) around each part of the find expression you want to remember

Then \n or \$n to get that value in the replacement, with n being the set of brackets in the sequence: the first is \1 or \$1 etc.

Further reading and resources

Mastering Regular Expressions, Jeffrey Friedl, O'Reilly

<http://pythex.org/> - enter some text and test a regex find (Python syntax)

<https://www.myregextester.com/> - do online find/replace

<http://regexper.com/> - display a regex as a railway diagram

<https://regex.alf.nu/> - regex golf, if you're really keen