Perl Regular Expression Quick Reference 1.05

N.B.: this quick reference is just that - some of the explanations have been simplified. For the authoritative documentation, see the latest edition of *Programming Perl* or `perldoc perlre`.

Specific characters:

`\t`	A tab character
`\n`	A newline character (OS neutral)
`\r`	A carriage return character
`\f`	A form feed character
`\c`***X***	Control character **CTRL-*X***
`\`***NNN***	Octal code for character ***NNN***

Metacharacters:

The following 12 characters need to be escaped with a backslash - "****" - because by default, they mean something special.

\ | ( ) [{ ^ $ * + ? .

`.`	Match any one character (except `\n`)
`|`	Alternation
`( )`	Group and capture
`[ ]`	Define character class
`\`	Modify the meaning of the next char.

Anchors:

`^`	Match at the beginning of a string (or line)
`$`	Match at the end of a string (or line)
`\b`	Match at a 'word' boundary
`\B`	Match at not a 'word' boundary

These are also known as *zero width assertions.*

Quantifiers:

These quantifiers apply to the preceding *atom.*

`*`	Match 0 or more times
`+`	Match 1 or more times
`?`	Match 0 or 1 times
`{`***N***`}`	Match exactly ***N*** times
`{`***N***`,}`	Match at least ***N*** times
`{`***N***`,`***M***`}`	Match at least ***N*** but not more than ***M*** times

By default, quantifiers are "*greedy*". They attempt to match as many characters as possible. In order to make them match as few characters as possible, follow them with a question mark "**?**".

Character class metacharacters:

`^`	If the first character of a class, negates that class
`-`	Unless first or last character of a class, used for a range

Character class shortcuts:

`\d`	`[0-9]`	A digit
`\D`	`[^0-9]`	A non-digit
`\s`	`[ \t\n\r\f]`	A whitespace char.
`\S`	`[^ \t\n\r\f]`	A non-whitespace char.
`\w`	`[a-zA-Z0-9_]`	A 'word' char.
`\W`	`[^a-zA-Z0-9_]`	A 'non-word' char.

These shortcuts can be used either on their own, or within a character class.

Metaquote & case translations:

`\Q`	Quote (de-meta) characters until `\E`
`\U`	Uppercase characters until `\E`
`\L`	Lowercase characters until `\E`

Special variables:

`$``	The characters to the left of the match
`$&`	The characters that matched
`$'`	The characters to the right of the match
`\`***N***	The characters captured by the ***N***th set of parentheses (if on the match side)
`$`***N***	The characters captured by the ***N***th set of parentheses (if not on the match side)

Modifiers:

These modifiers apply to the entire pattern

`/i`	Ignore case
`/g`	Match globally (all)
`/m`	Let `^` and `$` match next to embedded `\n`
`/s`	Let `.` match `\n`
`/x`	Ignore most whitespace and allow comments
`/e`	Evaluate right hand side of `s///` as an expression

All except `/e` apply to both `m//` and `s///` .

Binding operators:

`=~`	True if the regex matches
`!~`	True if the regex doesn't match