Perl 6 Cheatsheet

Suman
suman81765@gmail.com
Kathmandu, Nepal

July 20, 2017

1 Regular Expressions

`~~`	Smart match operator.
`$/`	Variable that contains the matched part of string.
`~$/`	Stringify the `$/` variable.
`.match`	Method invocation syntax.
`/ /`	Pattern is kept between a pair of slash delimiters.
`m/ /` or `m{ }`	Can use other delimiters except a pair of slash if we prefix the pattern with the letter"`m`"
`rx/ /` or `rx{ }`	signifies a pattern which can be stored in a variable
`.`	matches any character
`\w`	word character, matches one single alphanumeric character (alphabetical characters, digits and `_` character)
`\W`	any other character than `\w`
`\d`	digits
`\D`	non-digits
`\s`	any kind of whitespace, not just vertical whitespace.
`\S`	non-whitespace
`\n`	newline
`\N`	non-newline
`\h`	matches a single horizontal whitespace character.
`\H`	matches a single character that is not a horizontal whitespace character.

`\t`	matches a single tab character.
`\T`	matches a single character that is not a tab.
`\v`	matches a single vertical whitespace character.
`\V`	matches a single character that is not vertical whitespace.
`<[ ]>`	character class
`<foo>`	subrule
`<-[ ]>`	Negating character class
`^`	anchor representing beginning of the string
`^^`	start of line in multiline strings
`$`	anchor representing end of the string
`$$`	end of line in multiline strings
`<?before string>`	match that comes before the string
`<!before string>`	match that does not come before the string
`<?after string>`	match that comes after the string
`<!after string>`	match that does not come after the string
`<?{ }`	Code assertion which will match if code block returns a true value
`<!{ }`	Code assertion which will match unless the code block returns a true value
`||`	First match alternation
`|`	Longest match alternation
`( )`	Capturing
`$0, $1`	capture numbers, first and second items of the matched object in list context
`:i, :ignorecase`	Ignore upper or lower case
`:s,:sigspace`	adverb that makes whitespace significant in regex pattern
`:m`	ignore marks
`:g`	global
`:r`	ratchet
`<|w>`	match a word boundary
`<!|w>`	not match a word boundary
`<<`	matches a left word boundary.
`>>`	matches a right word boundary.

1.1 Predefined subrules

`<alnum>`	`\w`	'alpha' plus 'digit'
`<alpha>`	`<:L>`	Alphabetic characters
`<blank>`	`\h`	Horizontal whitespace
`<cntrl>`		Control characters
`<digit>`	`\d`	Decimal digits
`<graph>`		'alnum' plus 'punct'
`<lower>`	`<:Ll>`	Lowercase characters
`<print>`		'graph' plus 'space', but no 'cntrl'
`<punct>`		Punctuation and Symbols (only Punct beyond ASCII)
`<space>`	`\s`	Whitespace
`<upper>`	`<:Lu>`	Uppercase characters
`<|wb>`		Word Boundary (zero-width assertion)
`<ww>`		Within Word (zero-width assertion)
`<xdigit>`		Hexadecimal digit [0-9A-Fa-f]

1.2 Quantifiers

`+`	matching preceding character one or more times. Quantifiers bind tighter than concatenation, so `ab+` matches one a followed by one or more b's. This is different for quotes, so `'ab'+` matches the strings ab, abab, abababab etc.
`*`	matching preceding character zero or more times
`?`	matching preceding character zero or one match
`** min..max`	at least `min` and at most `max` times.
`%`	modified quantifier
`:`	prevent backtracking

Perl 6 is Unicode compliant.

Whitespace is usually not significant within regex patterns unless specified with `:s` or `:sigspace` adverb.