

Anchors	
^	Start of line +
\A	Start of string +
\$	End of line +
\Z	End of string +
\b	Word boundary +
\B	Not word boundary +
\<	Start of word
\>	End of word

Sample Patterns	
([A-Za-z0-9-]+)	Letters, numbers and hyphens
(\d{1,2}\V\d{1,2}\V\d{4})	Date (e.g. 21/3/2006)
([^\s]+(?:\.(jpg gif png)))\.\2)	jpg, gif or png image
(^[1-9]{1}\$ ^[1-4]{1}[0-9]{1}\$ ^50\$)	Any number from 1 to 50 inclusive
(#?([A-Fa-f0-9]){3}([A-Fa-f0-9]){3})?)	Valid hexadecimal colour code
((?=[*\d])(?=[*a-z])(?=[*A-Z]).{8,15})	8 to 15 character string with at least one upper case letter, one lower case letter, and one digit (useful for passwords).
(\w+@[a-zA-Z_]+?\.[a-zA-Z]{2,6})	Email addresses
(\<(/?[^\>]+)\>)	HTML Tags

Character Classes	
\c	Control character
\s	White space
\S	Not white space
\d	Digit
\D	Not digit
\w	Word
\W	Not word
\xhh	Hexadecimal character hh
\Oxxx	Octal character xxx

POSIX Character Classes	
[ :upper:]	Upper case letters
[ :lower:]	Lower case letters
[ :alpha:]	All letters
[ :alnum:]	Digits and letters
[ :digit:]	Digits
[ :xdigit:]	Hexadecimal digits
[ :punct:]	Punctuation
[ :blank:]	Space and tab
[ :space:]	Blank characters
[ :cntrl:]	Control characters
[ :graph:]	Printed characters
[ :print:]	Printed characters and spaces
[ :word:]	Digits, letters and underscore

Assertions	
?=	Lookahead assertion +
?!	Negative lookahead +
?<=	Lookbehind assertion +
?!= or ?<!	Negative lookbehind +
?>	Once-only Subexpression
?()	Condition [if then]
?()	Condition [if then else]
?#	Comment

Note	These patterns are intended for reference purposes and have not been extensively tested. Please use with caution and test thoroughly before use.
------	--

Quantifiers	
*	0 or more +
*?	0 or more, ungreedy +
+	1 or more +
+?	1 or more, ungreedy +
?	0 or 1 +
??	0 or 1, ungreedy +
{3}	Exactly 3 +
{3,}	3 or more +
{3,5}	3, 4 or 5 +
{3,5}?	3, 4 or 5, ungreedy +

Special Characters	
\	Escape Character +
\n	New line +
\r	Carriage return +
\t	Tab +
\v	Vertical tab +
\f	Form feed +
\a	Alarm
[\b]	Backspace
\e	Escape
\N{name}	Named Character

String Replacement (Backreferences)	
\$n	nth non-passive group
\$2	"xyz" in /^(abc(xyz))\$/
\$1	"xyz" in /^(?:abc)(xyz)\$/
\$`	Before matched string
\$'	After matched string
\$+	Last matched string
\$&	Entire matched string
\$_	Entire input string
\$\$	Literal "\$"

Ranges	
.	Any character except new line (\n) +
(a b)	a or b +
(...)	Group +
(?:...)	Passive Group +
[abc]	Range (a or b or c) +
[^abc]	Not a or b or c +
[a-q]	Letter between a and q +
[A-Q]	Upper case letter + between A and Q +
[0-7]	Digit between 0 and 7 +
\n	nth group/subpattern +

Note	Ranges are inclusive.
------	-----------------------

Pattern Modifiers	
g	Global match
i	Case-insensitive
m	Multiple lines
s	Treat string as single line
x	Allow comments and white space in pattern
e	Evaluate replacement
U	Ungreedy pattern

Metacharacters (must be escaped)		
^	[	.
\$	{	*
(	\	+
)		?
<	>	

Note	Items marked + should work in most regular expression implementations.
------	--

Available free from  
AddedBytes.com