# Anchors

|  |  |
| --- | --- |
| ^ | Start of string or line in multi-line pattern |
| \A | Start of string |
| $ | End of string or line in multi-line pattern |
| \Z | End of string |
| \b | Word boundary |
| \B | Not word boundary |
| \< | Start of word |
| \> | End of word |

# Character Classes

|  |  |
| --- | --- |
| \c | Control character |
| \s | White space |
| \S | Not white space |
| \d | Digit |
| \D | Not digit |
| \w | Word |
| \W | Not word |
| \x | Hexade­cimal digit |
| \O | Octal digit |

# POSIX

|  |  |
| --- | --- |
| [: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 |

# Quantifiers

|  |  |  |  |
| --- | --- | --- | --- |
| \* | 0 or more | {3} | Exactly 3 |
| + | 1 or more | {3,} | 3 or more |
| ? | 0 or 1 | {3,5} | 3, 4 or 5 |

Add a ? to a quantifier to make it non-greedy.

# Groups & Ranges

|  |  |
| --- | --- |
| . | Any character except new line (\n) |
| (a|b) | a or b |
| (...) | Group |
| (?:...) | Passive (non-c­apt­uring) group |
| [abc] | Range (a or b or c) |
| [^abc] | Not (a or b or c) |
| [a-q] | Lower case letter from a to q |
| [A-Q] | Upper case letter from A to Q |
| [0-7] | Digit from 0 to 7 |
| \x | Group/­sub­pattern number "­x" |

Ranges are inclusive.

# Pattern Modifiers

|  |  |
| --- | --- |
| g | Global match |
| i \* | Case-i­nse­nsitive |
| m \* | Multiple lines |
| s \* | Treat string as single line |
| x \* | Allow comments & whitespace in pattern |
| e \* | Evaluate replac­ement |
| U \* | Ungreedy pattern |

\* PCRE modifier

# String Replacement

|  |  |
| --- | --- |
| $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 |

Some regex implementations use \ instead of $

# Common Metacharacters

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

The escape character is usually \

# Special Characters

|  |  |
| --- | --- |
| \n | New line |
| \r | Carriage return |
| \t | Tab |
| \v | Vertical tab |
| \f | Form feed |
| \xxx | Octal character xxx |
| \xhh | Hex character hh |

# Helpful Resources

* [Regex Quickstart](https://www.rexegg.com/regex-quickstart.html)
* [Simple Regex Cheat Sheet](https://www.petefreitag.com/cheatsheets/regex/)
* [MIT Regex Cheat Sheet (PDF)](http://web.mit.edu/hackl/www/lab/turkshop/slides/regex-cheatsheet.pdf)
* [regex101: build, test, and debug regex](https://regex101.com/)