# RegularExpression Cheat Sheet and Quick Reference

## Special Characters

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

## Operators

| | |
|---|---|
| \| | Or |
| * | 0 or more times. Match as many times as possible. |
| + | 1 or more times. Match as many times as possible. |
| ? | 0 or 1 times. Prefer 1. |
| *? | 0 or more times. Match as few times as possible. |
| +? | 1 or more times. Match as few times as possible. |
| ?? | 0 or 1 times. Prefer 0. |
| *+ | 0 or more times. Match as many times as possible when first encountered, do not retry with fewer even if overall match fails (Possessive Match). |
| ++ | 1 or more times. Possessive match. |
| ?+ | 0 or 1 times. Possessive match. |
| {n}<br>{n}?<br>{n}+ | Exactly n times. |
| {n,} | n or more. |
| {n,}? | At least n times, but no more than required for an overall pattern match. |
| {n,m} | Between n and m times. |
| {n,m}? | Between n and m times. Match as few times as possible, but not less than n. |

## Anchors

| | |
|---|---|
| ^ | Beginning of a line. |
| $ | End of a line. |
| \A | Beginning of an input. Doesn't match after a new line within the input. |
| \z | End of input. |
| \Z | End of input, but before the final line terminator, if one exists. |
| . | Any character. |
| \ | Quote (escape) following character. |

## Others

| | |
|---|---|
| $n | n is a digit. Back referencing to a capture group. n must be >= 0 and not greater than the number of capture groups. $ not followed by a digit has no special meaning. |
| \ | Treat the following character as a literal, suppressing any special meaning. |

## Character Classes

| | |
|---|---|
| \b | Word boundary, if outside of a [Set].<br>BACKSPACE, if within a [Set]. |
| \B | Not word boundary. |
| \s | White space character. |
| \S | Non-white space character. |
| \d | Digit character. |
| \D | Non-digit character. |
| \w | Word character. |
| \W | Non-word character. |

## Groups and Ranges

| | |
|---|---|
| (...) | Capturing parentheses (capturing group). |
| (?:...) | Non-capturing parentheses. Matches but doesn't capture. Somewhat more efficient than capturing parentheses. |
| (?!...) | Negative look-ahead. True if the parenthesized pattern does not match at the current input position. |
| [...] | Any one character in the set. |
| [^...] | Negated set. Not any one in the set. |

## Useful Examples

| |
|---|
| `m[^o]` matches any "m" followed by anything other than "o" |
| `m(?!o)` matches any "m" (and only "m") not followed by "o" |
| `(?<=,|^)([^,]*)(,\1)+(?=,|$)` matches consecutive duplicates from a comma-delimited list[1] |
| `<([a-z][a-z0-9]*)\b[^>]*>(.*?)</\1>` matches any HTML or XML tags[1] |

[1] From http://www.regular-expressions.info/duplicatelines.html by Jan Goyvaerts.



Version 2.0. Copyright 2023 Kodeco, Inc. All rights reserved.