**RegEx**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Expression | Meaning | Example | Legal | Not Legal |
| Literal Characters |  | ABC | ABC  XYZABCQWE | DEF  CBA |
| ^ | Starting | ^ABC | ABC  ABCQWE | XYZABC |
| $ | Ending | ABC$ | ABC  XYZABC | ABC123 |
| ^ … $ | Starting + Ending | ^ABC$ | ABC | xABCy  xABC  ABCy |
| . | Any Single Char | ^AB.C$ | AB2C  ABxC | ABC  ABxyC |
| \* | Previous expression can be 0 or more times | ^AB\*C$ | ABBBBBC  ABC  AC | AB  AB8C |
| () | Creating a single expression from multiple small expressions | ^(AB)\*C$ | ABABABABC  C  ABC | ABDC  AC  BC  ABBC |
| + | Previous expression must be 1 or more times | ^AB+C$ | ABBC  ABBBBBBC  ABC | AC  ABDC |
| ? | Previous expression can be 0 or 1 times | ^AB?C$ | ABC  AC | ABBC |
| { n } | Previous expression must be exactly n times | ^AB{3}C$ | ABBBC | ABC  ABBC  ABBBBC |
| { n, m } | Previous expression can be n to m times | ^AB{3,5}C$ | ABBBC  ABBBBC  ABBBBBC | ABBC  ABC |
| { n, } | Previous expression must be minimum n times | ^AB{3,}C$ | ABBBC  ABBBBBBBBBC | ABBC  ABC |
| | | Or | ^AB|C$ | AB  AC | AD  A  B  C |
|  |  | ^(Dog|Cat|Fish)$ | Dog  Cat  Fish | DogCat  Puppy |
| [chars] | Only one of the given chars | ^AB[xyz123]C$ | ABxC  AByC  ABzC  AB1C  AB2C  AB3C | ABC  ABwC  ABxyC |
| [char-char] | Only one char in the given range | ^AB[e-n]C$ | ABfC  ABgC | ABaC |
|  |  | ^AB[4-8]C$ | AB7C  AB4C | AB1C |
| [^chars] | Only one char which is not in the given chars | ^AB[^xyz]C$ | ABwC | ABxC |
| [^chars-chars] | Only one char which is not in the given range | ^AB[^3-8]C$ | AB1C  ABwC | AB3C  AB4C |
| \ | Escaped Character – Turning expression char into literal char | ^AB\.C$ | AB.C | AB4C  ABwC |
| \d | Digit | ^AB\dC$ | AB7C | ABwC |
| \w | Digit / \_ / Letter | ^AB\wC$ | AB8C  ABtC  AB\_C | AB;C |
| \s | White character (space, tab, enter…) | ^AB\sC$ | AB C | AB3C |
|  | Escaped Character – Turning a literal char into regex expression char | ^AB[xyz\d]C$ | AB3C  ABxC | ABwC  ABdC |