

.NET FRAMEWORK REGULAR EXPRESSIONS



SINGLE CHARACTERS

| Use | To match any character |
|-------------------------|-----------------------------|
| [<i>set</i>] | In that set |
| [^ <i>set</i>] | Not in that set |
| [<i>a-z</i>] | In the <i>a-z</i> range |
| [^ <i>a-z</i>] | Not in the <i>a-z</i> range |
| . | Any except \n (new line) |
| \char | Escaped special character |

CONTROL CHARACTERS

| Use | To match | Unicode |
|----------------|-------------------------|---------|
| \t | Horizontal tab | \u0009 |
| \v | Vertical tab | \u000B |
| \b | Backspace | \u0008 |
| \e | Escape | \u001B |
| \r | Carriage return | \u000D |
| \f | Form feed | \u000C |
| \n | New line | \u000A |
| \a | Bell (alarm) | \u0007 |
| \c <i>char</i> | ASCII control character | — |

NON-ASCII CODES

| Use | To match character with |
|--------|--------------------------------|
| \octal | 2-3 digit octal character code |
| \x hex | 2-digit hex character code |
| \u hex | 4-digit hex character code |

CHARACTER CLASSES

| Use | To match character |
|--------------------|---------------------------------------|
| \p{ <i>ctgry</i> } | In that Unicode category or block |
| \P{ <i>ctgry</i> } | Not in that Unicode category or block |
| \w | Word character |
| \W | Non-word character |
| \d | Decimal digit |
| \D | Not a decimal digit |
| \s | White-space character |
| \S | Non-white-space char |

QUANTIFIERS

| Greedy | Lazy | Matches |
|-------------------------|--------------------------|---------------------------------|
| * | *? | 0 or more times |
| + | +? | 1 or more times |
| ? | ?? | 0 or 1 time |
| { <i>n</i> } | { <i>n</i> }? | Exactly <i>n</i> times |
| { <i>n</i> ,} | { <i>n</i> ,}? | At least <i>n</i> times |
| { <i>n</i> , <i>m</i> } | { <i>n</i> , <i>m</i> }? | From <i>n</i> to <i>m</i> times |

ANCHORS

| Use | To specify position |
|-----|--|
| ^ | At start of string or line |
| \A | At start of string |
| \z | At end of string |
| \Z | At end (or before \n at end) of string |
| \$ | At end (or before \n at end) of string or line |
| \G | Where previous match ended |
| \b | On word boundary |
| \B | Not on word boundary |

GROUPS

| Use | To define |
|------------------------------------|--------------------------------|
| (<i>exp</i>) | Indexed group |
| (? <i><name>exp</i>) | Named group |
| (? <i><name1-name2>exp</i>) | Balancing group |
| (?: <i>exp</i>) | Noncapturing group |
| (?= <i>exp</i>) | Zero-width positive lookahead |
| (?! <i>exp</i>) | Zero-width negative lookahead |
| (?<= <i>exp</i>) | Zero-width positive lookbehind |
| (?<! <i>exp</i>) | Zero-width negative lookbehind |
| (?> <i>exp</i>) | Non-backtracking (greedy) |

INLINE OPTIONS

| Option | Effect on match |
|--------|--------------------|
| i | Case-insensitive |
| m | Multiline mode |
| n | Explicit (named) |
| s | Single-line mode |
| x | Ignore white space |

| Use | To |
|-----------------------------|--|
| (? <i>imnsx-imnsx</i>) | Set or disable the specified options |
| (? <i>imnsx-imnsx:exp</i>) | Set or disable the specified options within the expression |

June 2014

© 2014 Microsoft. All rights reserved.

BACKREFERENCES

| Use | To match |
|-----------------------------|---------------|
| <code>\n</code> | Indexed group |
| <code>\k<name></code> | Named group |

ALTERNATION

| Use | To match |
|---|--|
| <code>a b</code> | Either <i>a</i> or <i>b</i> |
| <code>(?exp)</code> <i>yes no</i> | <i>yes</i> if <i>exp</i> is matched <i>no</i> if <i>exp</i> isn't matched |
| <code>(?name)</code> <i>yes no</i> | <i>yes</i> if <i>name</i> is matched <i>no</i> if <i>name</i> isn't matched |

SUBSTITUTION

| Use | To substitute |
|-----------------------|--|
| <code>\$n</code> | Substring matched by group number <i>n</i> |
| <code>\${name}</code> | Substring matched by group <i>name</i> |
| <code>\$\$</code> | Literal \$ character |
| <code>\$&</code> | Copy of whole match |
| <code>\$`</code> | Text before the match |
| <code>\$'</code> | Text after the match |
| <code>\$+</code> | Last captured group |
| <code>\$_</code> | Entire input string |

COMMENTS

| Use | To |
|---------------------------|--------------------|
| <code>(?# comment)</code> | Add inline comment |
| <code>#</code> | Add x-mode comment |

For detailed information and examples, see <http://aka.ms/regex>

To test your regular expressions, see <http://regexlib.com/RETester.aspx>

SUPPORTED UNICODE CATEGORIES

| Category | Description |
|-----------|---------------------------------|
| Lu | Letter, uppercase |
| Ll | Letter, lowercase |
| Lt | Letter, title case |
| Lm | Letter, modifier |
| Lo | Letter, other |
| L | Letter, all |
| Mn | Mark, nonspacing combining |
| Mc | Mark, spacing combining |
| Me | Mark, enclosing combining |
| M | Mark, all diacritic |
| Nd | Number, decimal digit |
| Nl | Number, letterlike |
| No | Number, other |
| N | Number, all |
| Pc | Punctuation, connector |
| Pd | Punctuation, dash |
| Ps | Punctuation, opening mark |
| Pe | Punctuation, closing mark |
| Pi | Punctuation, initial quote mark |
| Pf | Punctuation, final quote mark |
| Po | Punctuation, other |
| P | Punctuation, all |
| Sm | Symbol, math |
| Sc | Symbol, currency |
| Sk | Symbol, modifier |
| So | Symbol, other |
| S | Symbol, all |
| Zs | Separator, space |
| Zl | Separator, line |
| Zp | Separator, paragraph |
| Z | Separator, all |
| Cc | Control code |
| Cf | Format control character |
| Cs | Surrogate code point |
| Co | Private-use character |
| Cn | Unassigned |
| C | Control characters, all |

For named character set blocks (e.g., Cyrillic), search for "supported named blocks" in the MSDN Library.

REGULAR EXPRESSION OPERATIONS

Class: `System.Text.RegularExpressions.Regex`

Pattern matching with Regex objects

| To initialize with | Use constructor |
|--------------------|--|
| Regular exp | <code>Regex(String)</code> |
| + options | <code>Regex(String, RegexOptions)</code> |
| + time-out | <code>Regex(String, RegexOptions, TimeSpan)</code> |

Pattern matching with static methods

Use an overload of a method below to supply the regular expression and the text you want to search.

Finding and replacing matched patterns

| To | Use method |
|-----------------------|---|
| Validate match | <code>Regex.IsMatch</code> |
| Retrieve single match | <code>Regex.Match</code> (first) <code>Match.NextMatch</code> (next) |
| Retrieve all matches | <code>Regex.Matches</code> |
| Replace match | <code>Regex.Replace</code> |
| Divide text | <code>Regex.Split</code> |
| Handle char escapes | <code>Regex.Escape</code> <code>Regex.Unescape</code> |

Getting info about regular expression patterns

| To get | Use Regex API |
|---------------|---|
| Group names | <code>GetGroupNames</code> <code>GetGroupNameFromNumber</code> |
| Group numbers | <code>GetGroupNumbers</code> <code>GetGroupNumberFromName</code> |
| Expression | <code>ToString</code> |
| Options | <code>Options</code> |
| Time-out | <code>MatchTimeout</code> |
| Cache size | <code>CacheSize</code> |
| Direction | <code>RightToLeft</code> |

