

Expresiones regulares

Character classes

<code>[abc]</code>	a, b, or c (simple class)
<code>[^abc]</code>	Any character except a, b, or c (negation)
<code>[a-zA-Z]</code>	a through z or A through Z, inclusive (range)
<code>[a-d[m-p]]</code>	a through d, or m through p: <code>[a-dm-p]</code> (union)
<code>[a-z&&[def]]</code>	d, e, or f (intersection)
<code>[a-z&&[^bc]]</code>	a through z, except for b and c: <code>[ad-z]</code> (subtraction)
<code>[a-z&&[^m-p]]</code>	a through z, and not m through p: <code>[a-lq-z]</code> (subtraction)

Predefined character classes

<code>.</code>	Any character (may or may not match line terminators)
<code>\d</code>	A digit: <code>[0-9]</code>
<code>\D</code>	A non-digit: <code>[^0-9]</code>
<code>\s</code>	A whitespace character: <code>[\t\n\x0B\f\r]</code>
<code>\S</code>	A non-whitespace character: <code>[^\s]</code>
<code>\w</code>	A word character: <code>[a-zA-Z_0-9]</code>
<code>\W</code>	A non-word character: <code>[^\w]</code>

Reluctant quantifiers

<code>X??</code>	X , once or not at all
<code>X*?</code>	X , zero or more times
<code>X+?</code>	X , one or more times
<code>X{n}?</code>	X , exactly n times
<code>X{n,}?</code>	X , at least n times
<code>X{n,m}?</code>	X , at least n but not more than m times