Regular Expressions and RegExp Object

Regular Expressions provide pattern matching and control operators for extracting information from strings which may be retrieved for reuse. Regex expressions can also be evaluated using methods of the Javascript String Object like match, replace and search.

Each Javascript RegExp also has a scope defintion which controls the range of the regular expression to a single, multi-line or global (all lines).

Strings can also be directly executed by methods of the Regular Expression Object.

RegExp Definition

Regular Expressions accepts two(2) definition formats:

- RegExp expressions can be entered into a Javascript variable and must be bracketed by right slash (/) as shown below:

```
var regexp = /regular expression/,scope defintion;
```

- A RegExp can also be created using the RegExp Javascript Object. It has two(2) parameters:

  Syntax

```
var pattern = new RegExp(pattern, attributes);
              (or) simply
var pattern = /pattern/attributes;
```

Parameters

pattern A string that specifies the pattern of the regular expression or another regular expression.

attributes An optional string containing any of the "g", "i", and "m" attributes that specify global, case-insensitive, and multiline matches, respectively.

Brackets

Brackets ([]) have a special meaning when used in the context of regular expressions. They are used to find a range of characters.

Expression	Description
[...]	Any one character between the brackets.
[^...]	Any one character not between the brackets.
[0-9]	It matches any decimal digit from 0 through 9.
[a-z]	It matches any character from lowercase a through lowercase z.
[A-Z]	It matches any character from uppercase A through uppercase Z.
[a-Z]	It matches any character from lowercase a through uppercase Z.

Quantifiers

The frequency or position of bracketed character sequences and single characters can be denoted by a special character. Each pecial character having a specific connotation. The +, *, ?, and $ flags all follow a character sequence.

Expression	Description
p+	It matches any string containing at least one p.
p*	It matches any string containing zero or more p's.
p?	It matches any string containing one or more p's.
pN	It matches any string containing a sequence of N p's
p2,3	It matches any string containing a sequence of two or three p's.
p2,	It matches any string containing a sequence of at least two p's.
p$	It matches any string with p at the end of it.
^p	It matches any string with p at the beginning of it.

Examples

Following examples will clear your concepts about matching chracters.

[â-zA-Z] - It matches any string not containing any of the characters ranging from a through z and A through Z.

p.p - It matches any string containing p, followed by any character, in turn followed by another p.

^{2}$ - It matches any string containing exactly two characters.

$< b > (.*) < /b >$ - It matches any string enclosed within $< b > and < /b >$.

p(hp)* - It matches any string containing a p followed by zero or more instances of the sequence hp.

Literal characters

Character	Description
Alphanumeric	Itself
\0	The NUL character (\u0000)
\t	Tab (\u0009)
\n	Newline (\u000A)
\v	Vertical tab (\u000B)
\f	Form feed (\u000C)
\r	carriage return (\u000D)

Metacharacters

A metacharacter is simply an alphabetical character preceeded by a backslash that acts to give the combination a special meaning.

Following is the list of metacharacters which can be used in Regular Expressions.

Modifiers

Several modifiers are available that can make your work with regexps much easier, like case sensitivity, searching in multiple lines etc.

RegExp Methods

Character	Description
.	A single character
\s	A whitespace character (space, tab, newline)
\S	non-whitespace character
\d	A digit (0-9)
\D	A non-digit
\w	A word character (a-z, A-Z, 0-9, _)
\W	A non-word character
[\b]	A literal backspace (special case).
[aeiou]	Matches a single character in the given set
[âeiou]	Matches a single character outside the given set
(foo|bar|baz)	Matches any of the alternatives specified

Table 1: Metacharacters

Modifier	Description
i	Perform case-insensitive matching.
m	Specifies that if the string has newline or carriage return characters, the ând $ operators will now match against a newline boundary instead of a string boundary
g	Perform a global matchthat is, find all matches rather than stopping after the first match.

Table 2: Modifiers

Method	Description
exec()	Executes a search for a match in its string parameter.
test()	Tests for a match in its string parameter.
toSource()	Returns an object literal representing the specified object; you can use this value to create a new object.
toString()	Returns a string representing the specified object.

Table 3: RegExp Methods

