**­­RegExp Object**

A regular expression is an object that describes a pattern of characters.

Regular expressions are used to perform pattern-matching and "search-and-replace" functions on text.

**Syntax**

/*pattern*/*modifiers*;

**Example**

var patt = /hello/i

Example explained:

* **/hello/i**  is a regular expression.
* **hello** is a pattern (to be used in a search).
* **i**  is a modifier (modifies the search to be case-insensitive).

**Modifiers**

Modifiers are used to perform case-insensitive and global searches:

|  |  |
| --- | --- |
| **Modifier** | **Description** |
| [i](http://www.w3schools.com/jsref/jsref_regexp_i.asp) | Perform case-insensitive matching |
| [g](http://www.w3schools.com/jsref/jsref_regexp_g.asp) | Perform a global match (find all matches rather than stopping after the first match) |
| [m](http://www.w3schools.com/jsref/jsref_regexp_m.asp) | Perform multiline matching |

**Brackets**

Brackets are used to find a range of characters:

|  |  |
| --- | --- |
| **Expression** | **Description** |
| [[abc]](http://www.w3schools.com/jsref/jsref_regexp_charset.asp) | Find any character between the brackets |
| [[^abc]](http://www.w3schools.com/jsref/jsref_regexp_charset_not.asp) | Find any character NOT between the brackets |
| [[0-9]](http://www.w3schools.com/jsref/jsref_regexp_0-9.asp) | Find any digit between the brackets |
| [[^0-9]](http://www.w3schools.com/jsref/jsref_regexp_not_0-9.asp) | Find any digit NOT between the brackets |
| [(x|y)](http://www.w3schools.com/jsref/jsref_regexp_xy.asp) | Find any of the alternatives specified |

**Metacharacters**

Metacharacters are characters with a special meaning:

|  |  |
| --- | --- |
| **Metacharacter** | **Description** |
| [.](http://www.w3schools.com/jsref/jsref_regexp_dot.asp) | Find a single character, except newline or line terminator |
| [\w](http://www.w3schools.com/jsref/jsref_regexp_wordchar.asp) | Find a word character (a-z, A-Z, 0-9 and underscore) |
| [\W](http://www.w3schools.com/jsref/jsref_regexp_wordchar_non.asp) | Find a non-word character |
| [\d](http://www.w3schools.com/jsref/jsref_regexp_digit.asp) | Find a digit­ |
| [\D](http://www.w3schools.com/jsref/jsref_regexp_digit_non.asp) | Find a non-digit character |
| [\s](http://www.w3schools.com/jsref/jsref_regexp_whitespace.asp) | Find a whitespace character |
| [\S](http://www.w3schools.com/jsref/jsref_regexp_whitespace_non.asp) | Find a non-whitespace character |
| [\b](http://www.w3schools.com/jsref/jsref_regexp_begin.asp) | Find a match at the beginning/end of a word |
| [\B](http://www.w3schools.com/jsref/jsref_regexp_begin_not.asp) | Find a match not at the beginning/end of a word |
| [\0](http://www.w3schools.com/jsref/jsref_regexp_nul.asp) | Find a NUL character |
| [\n](http://www.w3schools.com/jsref/jsref_regexp_newline.asp) | Find a new line character |
| [\f](http://www.w3schools.com/jsref/jsref_regexp_formfeed.asp) | Find a form feed character |
| [\r](http://www.w3schools.com/jsref/jsref_regexp_carriagereturn.asp) | Find a carriage return character |
| [\t](http://www.w3schools.com/jsref/jsref_regexp_tab.asp) | Find a tab character |
| [\v](http://www.w3schools.com/jsref/jsref_regexp_vtab.asp) | Find a vertical tab character |
|  |  |

**Regular Expression for E-Mail Validation** [**abc.xy1 @yahoo.co.in**](mailto:abc.xy1.lmn@yahoo.co.in)

var emailExp = /^\w+([\.-]?\w+)\*@\w+([\.]?\w+)\*(\.-\w{2,3})+$/;

* The two forward-slashes /.../ contains a regular expression.
* The leading ^ and trailing $ match the beginning and the ending of the input string, respectively.
* \w+ matches 1 or more word characters (a-z, A-Z, 0-9 and underscore).
* [.-] matches character . or -. The \ is known as the escape code, which restore the original literal meaning of the following character.
* [.-]? matches 0 or 1 occurrence of [.-].
* Again, \w+ matches 1 or more word characters.
* \* matches 0 or more occurrences of ([.-]?\w+).
* The sub-expression \w+([.-]?\w+)\* is used to match the username in the email, before the @ sign. It begins with at least one word character (a-z, A-Z, 0-9 and underscore), followed by more word characters or . or -. However, a . or - must follow by a word character (a-z, A-Z, 0-9 and underscore).
* The @ matches itself.
* Again, the sub-expression \w+([.-]?\w+)\* is used to match the email domain name, with the same pattern as the username described above.
* The sub-expression .\w{2,3} matches a . followed by two or three word characters, e.g., ".com", ".edu", ".us", ".uk", ".co".
* (.\w{2,3})+ specifies that the above sub-expression shall occur one or more times, e.g., ".co.uk", ".edu.sg" etc.