

## 26. JavaScript RegExp Object

RegExp, is short for regular expression.

### 26.1. What is RegExp?

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

When you search in a text, you can use a pattern to describe what you are searching for.

A simple pattern can be one single character.

A more complicated pattern can consist of more characters, and can be used for parsing, format checking, substitution and more.

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

### Syntax

```
var patt=new RegExp(pattern,modifiers);
```

or more simply:

```
var patt=/pattern/modifiers;
```

- pattern specifies the pattern of an expression
- modifiers specify if a search should be global, case-sensitive, etc.

### 26.2. RegExp Modifiers

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

The i modifier is used to perform case-insensitive matching.

The g modifier is used to perform a global match (find all matches rather than stopping after the first match).

### Example 1

Do a case-insensitive search for "iesmanacor" in a string:

```
var str="Visit IESManacor";  
var patt1=/iesmanacor/i;
```

The marked text below shows where the expression gets a match:

Visit IESManacor

## Example 2

Do a global search for "is":

```
var str="Is this all there is?";  
var patt1=/is/g;
```

The marked text below shows where the expression gets a match:

Is this all there is?

## Example 3

Do a global, case-insensitive search for "is":

```
var str="Is this all there is?";  
var patt1=/is/gi;
```

The marked text below shows where the expression gets a match:

Is this all there is?

## 26.3. test()

The test() method searches a string for a specified value, and returns true or false, depending on the result.

The following example searches a string for the character "e":

## Example

```
var patt1=new RegExp("e");  
document.write(patt1.test("The best things in life are  
free"));
```

Since there is an "e" in the string, the output of the code above will be:

```
true
```

## 26.4. exec()

The `exec()` method searches a string for a specified value, and returns the text of the found value. If no match is found, it returns *null*.

The following example searches a string for the character "e":

### Example 1

```
var patt1=new RegExp("e");  
document.write(patt1.exec("The best things in life  
are free"));
```

Since there is an "e" in the string, the output of the code above will be:

```
e
```

## 26.5. Complete RegExp Object Reference

For a complete reference of all the properties and methods that can be used with the RegExp object, go to our complete RegExp object reference.

The reference contains a brief description of use for each property and method!

### Modifiers

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

Modifier	Description
i	Perform case-insensitive matching
g	Perform a global match (find all matches rather than stopping after the first match)
m	Perform multiline matching

### Brackets

Brackets are used to find a range of characters:

Expression	Description
[abc]	Find any character between the brackets
[^abc]	Find any character not between the brackets
[0-9]	Find any digit from 0 to 9
[A-Z]	Find any character from uppercase A to uppercase Z
[a-z]	Find any character from lowercase a to lowercase z
[A-z]	Find any character from uppercase A to lowercase z
[adgk]	Find any character in the given set
[^adgk]	Find any character outside the given set
(red blue green)	Find any of the alternatives specified

## Metacharacters

Metacharacters are characters with a special meaning:

Metacharacter	Description
.	Find a single character, except newline or line terminator
\w	Find a word character
\W	Find a non-word character
\d	Find a digit
\D	Find a non-digit character
\s	Find a whitespace character
\S	Find a non-whitespace character
\b	Find a match at the beginning/end of a word
\B	Find a match not at the beginning/end of a word
\0	Find a NUL character
\n	Find a new line character
\f	Find a form feed character

\r	Find a carriage return character
\t	Find a tab character
\v	Find a vertical tab character
\xxx	Find the character specified by an octal number xxx
\xdd	Find the character specified by a hexadecimal number dd
\uxxxx	Find the Unicode character specified by a hexadecimal number xxxx

## Quantifiers

Quantifier	Description
n+	Matches any string that contains at least one n
n*	Matches any string that contains zero or more occurrences of n
n?	Matches any string that contains zero or one occurrences of n
n{X}	Matches any string that contains a sequence of X n's
n{X,Y}	Matches any string that contains a sequence of X to Y n's
n{X,}	Matches any string that contains a sequence of at least X n's
n\$	Matches any string with n at the end of it
^n	Matches any string with n at the beginning of it
?=n	Matches any string that is followed by a specific string n
?!n	Matches any string that is not followed by a specific string n

## RegExp Object Properties

Property	Description
global	Specifies if the "g" modifier is set
ignoreCase	Specifies if the "i" modifier is set
lastIndex	The index at which to start the next match

multiline	Specifies if the "m" modifier is set
source	The text of the RegExp pattern

## RegExp Object Methods

Method	Description
compile()	Compiles a regular expression
exec()	Tests for a match in a string. Returns the first match
test()	Tests for a match in a string. Returns true or false