A regular expression is a sequence of characters that forms a search pattern.

The search pattern can be used for text search and text replace operations.

**What Is a Regular Expression?**

A regular expression is a sequence of characters that forms a **search pattern**.

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

A regular expression can be a single character, or a more complicated pattern.

Regular expressions can be used to perform all types of **text search** and **text replace** operations.

Syntax

/*pattern*/*modifiers*;

### **Example**

var patt = /bepractical/i;

Example explained:

**/** **bepractical /i**  is a regular expression.

**bepractical** is a pattern (to be used in a search).

**i**  is a modifier (modifies the search to be case-insensitive).

Example

<!DOCTYPE html>

<html>

<body>

<p>Search a string for "Be-practical", and display the position of the match:</p>

<button onclick="myFunction()">Try it</button>

<p id="demo"></p>

<script>

function myFunction() {

var str = "Visit Be-practical!";

var n = str.search(/Be-practical /i);

document.getElementById("demo").innerHTML = n;

}

</script>

</body>

</html>

## Modifiers

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

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

**Brackets**

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

Brackets are used to find a range of characters:

**Metacharacters**

Metacharacters are characters with a special meaning:

|  |  |
| --- | --- |
| **Metacharacter** | **Description** |
| [.](https://www.w3schools.com/jsref/jsref_regexp_dot.asp) | Find a single character, except newline or line terminator |
| [\w](https://www.w3schools.com/jsref/jsref_regexp_wordchar.asp) | Find a word character |
| [\W](https://www.w3schools.com/jsref/jsref_regexp_wordchar_non.asp) | Find a non-word character |
| [\d](https://www.w3schools.com/jsref/jsref_regexp_digit.asp) | Find a digit |
| [\D](https://www.w3schools.com/jsref/jsref_regexp_digit_non.asp) | Find a non-digit character |
| [\s](https://www.w3schools.com/jsref/jsref_regexp_whitespace.asp) | Find a whitespace character |
| [\S](https://www.w3schools.com/jsref/jsref_regexp_whitespace_non.asp) | Find a non-whitespace character |
| [\b](https://www.w3schools.com/jsref/jsref_regexp_begin.asp) | Find a match at the beginning/end of a word |
| [\B](https://www.w3schools.com/jsref/jsref_regexp_begin_not.asp) | Find a match not at the beginning/end of a word |
| [\0](https://www.w3schools.com/jsref/jsref_regexp_nul.asp) | Find a NUL character |
| [\n](https://www.w3schools.com/jsref/jsref_regexp_newline.asp) | Find a new line character |
| [\f](https://www.w3schools.com/jsref/jsref_regexp_formfeed.asp) | Find a form feed character |
| [\r](https://www.w3schools.com/jsref/jsref_regexp_carriagereturn.asp) | Find a carriage return character |
| [\t](https://www.w3schools.com/jsref/jsref_regexp_tab.asp) | Find a tab character |
| [\v](https://www.w3schools.com/jsref/jsref_regexp_vtab.asp) | Find a vertical tab character |
| [\xxx](https://www.w3schools.com/jsref/jsref_regexp_octal.asp) | Find the character specified by an octal number xxx |
| [\xdd](https://www.w3schools.com/jsref/jsref_regexp_hex.asp) | Find the character specified by a hexadecimal number dd |
| [\uxxxx](https://www.w3schools.com/jsref/jsref_regexp_unicode_hex.asp) | Find the Unicode character specified by a hexadecimal number xxxx |

**Quantifiers**

|  |  |
| --- | --- |
| **Quantifier** | **Description** |
| [n+](https://www.w3schools.com/jsref/jsref_regexp_onemore.asp) | Matches any string that contains at least one *n* |
| [n\*](https://www.w3schools.com/jsref/jsref_regexp_zeromore.asp) | Matches any string that contains zero or more occurrences of *n* |
| [n?](https://www.w3schools.com/jsref/jsref_regexp_zeroone.asp) | Matches any string that contains zero or one occurrences of *n* |
| [n{X}](https://www.w3schools.com/jsref/jsref_regexp_nx.asp) | Matches any string that contains a sequence of *X* *n*'s |
| [n{X,Y}](https://www.w3schools.com/jsref/jsref_regexp_nxy.asp) | Matches any string that contains a sequence of X to Y *n*'s |
| [n{X,}](https://www.w3schools.com/jsref/jsref_regexp_nxcomma.asp) | Matches any string that contains a sequence of at least X *n*'s |
| [n$](https://www.w3schools.com/jsref/jsref_regexp_ndollar.asp) | Matches any string with *n* at the end of it |
| [^n](https://www.w3schools.com/jsref/jsref_regexp_ncaret.asp) | Matches any string with *n* at the beginning of it |
| [?=n](https://www.w3schools.com/jsref/jsref_regexp_nfollow.asp) | Matches any string that is followed by a specific string *n* |
| [?!n](https://www.w3schools.com/jsref/jsref_regexp_nfollow_not.asp) | Matches any string that is not followed by a specific string *n* |

**RegExp Object Properties**

|  |  |
| --- | --- |
| **Property** | **Description** |
| [constructor](https://www.w3schools.com/jsref/jsref_regexp_constructor.asp) | Returns the function that created the RegExp object's prototype |
| [global](https://www.w3schools.com/jsref/jsref_regexp_global.asp) | Checks whether the "g" modifier is set |
| [ignoreCase](https://www.w3schools.com/jsref/jsref_regexp_ignorecase.asp) | Checks whether the "i" modifier is set |
| [lastIndex](https://www.w3schools.com/jsref/jsref_regexp_lastindex.asp) | Specifies the index at which to start the next match |
| [multiline](https://www.w3schools.com/jsref/jsref_regexp_multiline.asp) | Checks whether the "m" modifier is set |
| [source](https://www.w3schools.com/jsref/jsref_regexp_source.asp) | Returns the text of the RegExp pattern |

**RegExp Object Methods**

|  |  |
| --- | --- |
| **Method** | **Description** |
| [compile()](https://www.w3schools.com/jsref/jsref_regexp_compile.asp) | Deprecated in version 1.5. Compiles a regular expression |
| [exec()](https://www.w3schools.com/jsref/jsref_regexp_exec.asp) | Tests for a match in a string. Returns the first match |
| [test()](https://www.w3schools.com/jsref/jsref_regexp_test.asp) | Tests for a match in a string. Returns true or false |
| [toString()](https://www.w3schools.com/jsref/jsref_regexp_tostring.asp) | Returns the string value of the regular expression |

**Explanation of the said Regular expression (email id)**

Regular Expression Pattern

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

|  |  |
| --- | --- |
| **Character** | **Description** |
| / .. / | All regular expressions start and end with forward slashes. |
| ^ | Matches the beginning of the string or line. |
| \w+ | Matches one or more word characters including the underscore. Equivalent to [A-Za-z0-9\_]. |
| [\.-] | \ Indicates that the next character is special and not to be interpreted literally. .- matches character . or -. |
| ? | Matches the previous character 0 or 1 time. Here previous character is [.-]. |
| \w+ | Matches 1 or more word characters including the underscore. Equivalent to [A-Za-z0-9\_]. |
| \* | Matches the previous character 0 or more times. |
| ([.-]?\w+)\* | Matches 0 or more occurrences of [.-]?\w+. |
| \w+([.-]?\w+)\* | The sub-expression \w+([.-]?\w+)\* is used to match the username in the email. It begins with at least one or more word characters including the underscore, equivalent to [A-Za-z0-9\_]. , followed by . or - and . or - must follow by a word character (A-Za-z0-9\_). |
| @ | It matches only @ character. |
| \w+([.-]?\w+)\* | It matches the domain name with the same pattern of user name described above.. |
| \.\w{2,3} | It matches a . followed by two or three word characters, e.g., .edu, .org, .com, .uk, .us, .co etc. |
| + | The + sign specifies that the above sub-expression shall occur one or more times, e.g., .com, .co.us, .edu.uk etc. |
| $ | Matches the end of the string or line. |

## HTML Code

[view plaincopy to clipboardprint?](https://www.w3resource.com/javascript/form/email-validation.php)

1. <!DOCTYPE html**>**
2. **<html** lang="en"**>**
3. **<head>**
4. **<meta** charset="utf-8"**>**
5. **<title>**JavaScript form validation - checking email**</title>**
6. **<link** rel='stylesheet' href='form-style.css' type='text/css' **/>**
7. **</head>**
8. **<body** onload='document.form1.text1.focus()'**>**
9. **<div** class="mail"**>**
10. **<h2>**Input an email and Submit**</h2>**
11. **<form** name="form1" action="#"**>**
12. **<ul>**
13. **<li><input** type='text' name='text1'**/></li>**
14. **<li>**&nbsp;**</li>**
15. **<li** class="submit"**><input** type="submit" name="submit" value="Submit" onclick="ValidateEmail(document.form1.text1)"**/></li>**
16. **<li>**&nbsp;**</li>**
17. **</ul>**
18. **</form>**
19. **</div>**
20. **<script** src="email-validation.js"**></script>**
21. **</body>**
22. **</html>**

## JavaScript Code

[view plaincopy to clipboardprint?](https://www.w3resource.com/javascript/form/email-validation.php)

1. **function** ValidateEmail(inputText)
2. {
3. **var** mailformat = /^\w+([\.-]?\w+)\*@\w+([\.-]?\w+)\*(\.\w{2,3})+$/;
4. **if**(inputText.value.match(mailformat))
5. {
6. document.form1.text1.focus();
7. **return** **true**;
8. }
9. **else**
10. {
11. alert("You have entered an invalid email address!");
12. document.form1.text1.focus();
13. **return** **false**;
14. }
15. }

## CSS Code

[view plaincopy to clipboardprint?](https://www.w3resource.com/javascript/form/email-validation.php)

1. li {**list-style-type**: none;
2. **font-size**: 16pt;
3. }
4. .mail {
5. **margin**: auto;
6. **padding-top**: 10px;
7. **padding-bottom**: 10px;
8. **width**: 400px;
9. background : #D8F1F8;
10. **border**: 1px soild silver;
11. }
12. .mail h2 {
13. **margin-left**: 38px;
14. }
15. input {
16. **font-size**: 20pt;
17. }
18. input:focus, textarea:focus{
19. **background-color**: lightyellow;
20. }
21. input submit {
22. **font-size**: 12pt;
23. }
24. .rq {
25. **color**: #FF0000;
26. **font-size**: 10pt;
27. }

This is second Program Using the code

Here I will show you how to validate the HTML controls using JavaScript.

Take a text input in html and a button input like this

<input type='text' id='txtEmail'/>

<input type='submit' name='submit' onclick='Javascript:checkEmail();'/>

Now when the button is clicked then the JavaScript function SubmitFunction() will be called. Now write the bellow code in this function.

script language="javascript">

function checkEmail() {

var email = document.getElementById('txtEmail');

var filter = /^([a-zA-Z0-9\_\.\-])+\@(([a-zA-Z0-9\-])+\.)+([a-zA-Z0-9]{2,4})+$/;

if (!filter.test(email.value)) {

alert('Please provide a valid email address');

email.focus;

return false;

}

}</script>