

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <meta charset="utf-8">
5   <title>Using JavaScript String Functions</title>
6 <!--
7   cssdweb.edu/StudentResources/CodeSamples/UsingJavaScriptStringFunctions_1.html
8   Copyright (c) 2022 by Jesse Heines. All rights reserved. May be freely
9   copied or excerpted for educational purposes with credit to the author.
10  updated by JMH on May 18, 2022 at 5:58 PM
11 -->
12 <style>
13   p, blockquote, ol {
14     margin-top: 0 ;
15     margin-bottom: 0 ;
16   }
17   pre, code {
18     margin-bottom: 0 ;
19     font-size: 1.1em ;
20     font-style: normal ;
21   }
22 </style>
23 </head>
24
25 <body>
26   <h3>JavaScript String Functions Demonstration</h3>
27
28   <pre>(1) var str="New Hampshire Dept. of Corrections Special School District" ;</pre>
29   <p>Original string:&nbsp;<strong>
30   <script>
31     var str="New Hampshire Dept. of Corrections Special School District" ;
32     document.writeln( str ) ;
33   </script>
34   </strong></p>
35
36   <pre>(2) var length1 = str.length ;</pre>
37   <p>The original string contains <strong>
38   <script>
39     var length1 = str.length ;
40     document.writeln( length1 ) ;
41   </script>
42   </strong> characters.</p>
43   <blockquote><em>Note that the string length (character count) includes spaces.&nbsp;<br>
44     Also note that <code>length</code> is a <strong>property</strong>, not a
45     function.&nbsp;<br> This is why <code>length</code> is not followed by
46     parentheses.</em></blockquote>
47
48   <pre>(3) var space1 = str.indexOf( " " ) ;</pre>
49   <p>The first space is at character position:&nbsp;<strong>
50   <script>
51     var space1 = str.indexOf( " " ) ;
52     document.writeln( space1 ) ;
53   </script>
54   </strong></p>
55   <blockquote><em>Note that the character position is 0-based!</em></blockquote>
56
57   <pre>(4) var word1 = str.substr( 0, space1 ) ;</pre>
58   <p>The first word is:&nbsp;<strong>
59   <script>
60     var word1 = str.substr( 0, space1 ) ;
61     document.writeln( word1 ) ;
62   </script>
63   </strong></p>
```

```

64  <blockquote><em>The</em> <code>substr</code> <em>parameters are the starting position
65    and the number of characters.</em></blockquote>
66
67  <pre>(5) var space2 = str.indexOf( " ", space1+1 ) ;</pre>
68  <p>The second space is at character position:&nbsp;<strong>
69  <script>
70    var space2 = str.indexOf( " ", space1+1 ) ;
71    document.writeln( space2 ) ;
72  </script>
73  </strong></p>
74  <blockquote><em>The second parameter tells JavaScript where to start the character
75    search.</em></blockquote>
76
77  <pre>(6) var word2 = str.substr( space1+1, space2-space1 ) ;</pre>
78  <p>The second word &mdash; extracted using the <code>substr</code> function
79    &mdash; is:&nbsp;<strong>
80  <script>
81    var word2 = str.substr( space1+1, space2-space1 ) ;
82    document.writeln( word2 ) ;
83  </script>
84  </strong></p>
85  <blockquote><em>Remember that the second parameter to the</em> <code>substr</code>
86    <em>function is the <strong>number of characters</strong>, not the ending space.&nbsp;<strong>
87    This is why we have to subtract</em> <code>space1</code> <em>from</em>
88    <code>space2</code>.</blockquote>
89
90  <pre>(7) var word2b = str.substring( space1+1, space2 ) ;</pre>
91  <p>The second word &mdash; extracted using the <code>substring</code> function &mdash;
92    is:&nbsp;<strong>
93  <script>
94    var word2b = str.substring( space1+1, space2 ) ;
95    document.writeln( word2b ) ;
96  </script>
97  </strong></p>
98  <blockquote><em>The second parameter to the</em> <code>substring</code> <em>function is
99    indeed the ending space.&nbsp;<strong> This is why we do <strong>not</strong> subtract</em>
100   <code>space1</code> <em>from</em> <code>space2</code>.</blockquote>
101
102  <pre>(8) var phrase1 = word2 + " " + word1 ;</pre>
103  <p>Concatenating <code>word1</code> onto <code>word2</code> yields:&nbsp;<strong>
104  <script>
105    var phrase1 = word2 + " " + word1 ;
106    document.writeln( phrase1 ) ;
107  </script>
108  </strong></p>
109  <blockquote><em>Remember that the second parameter is the number of characters, not the
110    ending space.&nbsp;<strong> This is why we have to subtract</em> <code>space1</code>
111    <em>from</em> <code>space2</code>.</blockquote>
112
113  <pre>(9) var arrWords = str.split( " " ) ;</pre>
114  <p>We can get all the words at once using the <code>split</code> function:</p>
115  <strong>
116  <script>
117    var arrWords = str.split( " " ) ;
118    document.writeln( "<ol start='0'>" ) ;
119    for ( var k = 0 ; k < arrWords.length ; k++ ) {
120      document.writeln( "<li>" + arrWords[k] + "</li>" ) ;
121    }
122    document.writeln( "</ol>" ) ;
123  </script>
124  </strong>
125  <blockquote><em>Note that array indexes are 0-based!</em></blockquote>
126
127  <pre>(10) document.writeln( arrWords[5] + " " + arrWords[7] + " " + arrWords[3] + " " +

```

```

128     arrWords[1] + " " + arrWords[0] + " " + arrWords[6] + " " + arrWords[4] ) ;</pre>
129 <p>We can then print them any way we want!&nbsp;<strong></p>
130 <script>
131     document.writeln( "&nbsp; &nbsp; &nbsp; &nbsp; " +
132         arrWords[5] + " " + arrWords[7] + " " + arrWords[3] + " " +
133         arrWords[1] + " " + arrWords[0] + " " + arrWords[6] + " " + arrWords[4] ) ;
134 </script>
135 </strong>
136
137 <pre>(11) var phrase2 = str.replaceAll( " ", "|" ) ;</pre>
138 <p>Replacing all spaces with vertical bars yields:&nbsp;<strong></p>
139 <script>
140     var phrase2 = str.replaceAll( " ", "|" ) ;
141     document.writeln( phrase2 ) ;
142 </script>
143 </strong></p>
144 <blockquote><em>The first parameter is the search string and the second is the
145     replacement string.</em></blockquote>
146
147 <pre>(12) var bStartsWithNew = str.startsWith( "New" ) ;</pre>
148 <p>Testing whether the string starts with <strong>New</strong> yields:&nbsp;<strong></p>
149 <script>
150     var bStartsWithNew = str.startsWith( "New" ) ;
151     document.writeln( bStartsWithNew ) ;
152 </script>
153 </strong></p>
154
155 <pre>(13) var bEndsWithNew = str.startsWith( "Hampshire" ) ;</pre>
156 <p>Testing whether the string ends with <strong>Hampshire</strong> yields:&nbsp;<strong></p>
157 <script>
158     var bEndsWithNew = str.startsWith( "Hampshire" ) ;
159     document.writeln( bEndsWithNew ) ;
160 </script>
161 </strong></p>
162
163 <pre>(14) var strUpper = str.toUpperCase() ;</pre>
164 <p>Converting the string to all uppercase yields:&nbsp;<strong></p>
165 <script>
166     var strUpper = str.toUpperCase() ;
167     document.writeln( strUpper ) ;
168 </script>
169 </strong></p>
170
171 <pre>(15) var strLower = str.toLowerCase() ;</pre>
172 <p>Converting the string to all lowercase yields:&nbsp;<strong></p>
173 <script>
174     var strLower = str.toLowerCase() ;
175     document.writeln( strLower ) ;
176 </script>
177 </strong></p>
178
179 <pre>(16) var bIncludesSchool1 = str.includes( "School" ) ;</pre>
180 <p>Testing whether the string includes (contains) <strong>School</strong>
181     yields:&nbsp;<strong></p>
182 <script>
183     var bIncludesSchool1 = str.includes( "School" ) ;
184     document.writeln( bIncludesSchool1 ) ;
185 </script>
186 </strong></p>
187 <blockquote><em>Note that the <code>includes</code> function is case-sensitive.&nbsp;<strong>
188     This is verified by the next example.</em></blockquote>
189
190 <pre>(17) var bIncludesSchool2 = str.includes( "school" ) ;</pre>
191 <p>Testing whether the string includes <strong>school</strong> yields:&nbsp;<strong></p>

```

```
192 <script>
193     var bIncludesSchool2 = str.includes( "school" ) ;
194     document.writeln( bIncludesSchool2 ) ;
195 </script>
196 </strong></p>
197 </body>
198 </html>
```

JavaScript String Functions Demonstration

(1) `var str="New Hampshire Dept. of Corrections Special School District" ;`
Original string: **New Hampshire Dept. of Corrections Special School District**

(2) `var length1 = str.length ;`

The original string contains **58** characters.

*Note that the string length (character count) includes spaces. Also note that length is a **property**, not a function. This is why length is not followed by parentheses.*

(3) `var space1 = str.indexOf(" ") ;`

The first space is at character position: **3**

Note that the character position is 0-based!

(4) `var word1 = str.substr(0, space1) ;`

The first word is: **New**

The substr parameters are the starting position and the number of characters.

(5) `var space2 = str.indexOf(" ", space1+1) ;`

The second space is at character position: **13**

The second parameter tells JavaScript where to start the character search.

(6) `var word2 = str.substr(space1+1, space2-space1) ;`

The second word — extracted using the substr function — is: **Hampshire**

*Remember that the second parameter to the substr function is the **number of characters**, not the ending space. This is why we have to subtract space1 from space2.*

(7) `var word2b = str.substring(space1+1, space2) ;`

The second word — extracted using the substring function — is: **Hampshire**

*The second parameter to the substring function is indeed the ending space. This is why we do **not** subtract space1 from space2.*

(8) `var phrase1 = word2 + " " + word1 ;`

Concatenating word1 onto word2 yields: **Hampshire New**

Remember that the second parameter is the number of characters, not the ending space. This is why we have to subtract space1 from space2.

(9) `var arrWords = str.split(" ") ;`

We can get all the words at once using the split function:

0. New

1. Hampshire

2. Dept.

3. of

4. Corrections

5. Special

6. School

7. District

Note that array indexes are 0-based!

(10) `document.writeln(arrWords[5] + " " + arrWords[7] + " " + arrWords[3] + " " + arrWords[1] + " " + arrWords[0] + " " + arrWords[6] + " " + arrWords[4]) ;`

We can then print them any way we want!

Special District of Hampshire New School Corrections

```
(11) var phrase2 = str.replaceAll( " ", "|" ) ;
```

Replacing all spaces with vertical bars yields:

```
New|Hampshire|Dept.|of|Corrections|Special|School|District
```

The first parameter is the search string and the second is the replacement string.

```
(12) var bStartsWithNew = str.startsWith( "New" ) ;
```

Testing whether the string starts with **New** yields: **true**

```
(13) var bEndsWithNew = str.startsWith( "Hampshire" ) ;
```

Testing whether the string ends with **Hampshire** yields: **false**

```
(14) var strUpper = str.toUpperCase() ;
```

Converting the string to all uppercase yields: **NEW HAMPSHIRE DEPT. OF CORRECTIONS
SPECIAL SCHOOL DISTRICT**

```
(15) var strLower = str.toLowerCase() ;
```

Converting the string to all lowercase yields: **new hampshire dept. of corrections special school district**

```
(16) var bIncludesSchool1 = str.includes( "School" ) ;
```

Testing whether the string includes (contains) **School** yields: **true**

Note that the includes function is case-sensitive. This is verified by the next example.

```
(17) var bIncludesSchool2 = str.includes( "school" ) ;
```

Testing whether the string includes **school** yields: **false**