

Original Array	Sorted Alphabetically	Sorted Numerically
generated using the <code>Math.random()</code> function	sorted by simply calling the <code>.sort()</code> function	sorted by calling <code>.sort(cmp)</code> , where <code>cmp</code> is a comparison function
0. 77 1. 68 2. 42 3. 41 4. 6 5. 85 6. 80 7. 8 8. 35 9. 74 10. 33 11. 19 12. 0 13. 45 14. 87 15. 13 16. 23 17. 99 18. 41 19. 7	0. 0 1. 13 2. 19 3. 23 4. 33 5. 35 6. 41 7. 41 8. 42 9. 45 10. 6 11. 68 12. 7 13. 74 14. 77 15. 8 16. 80 17. 85 18. 87 19. 99	0. 0 1. 6 2. 7 3. 8 4. 13 5. 19 6. 23 7. 33 8. 35 9. 41 10. 41 11. 42 12. 45 13. 68 14. 74 15. 77 16. 80 17. 85 18. 87 19. 99

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4   <meta charset="utf-8">
5   <titleArray Example5 - Simple Sort</title>
6
7 <!--
8   File: \\cssd.Local\Student Resources\Website Development I\CodeSamples\ArrayExample_5_SimpleSort.html
9   Copyright (c) 2022 by Jesse Heines. All rights reserved. May be freely
10  copied or excerpted for educational purposes with credit to the author.
11  updated by JMH on June 7, 2022 at 11:32 PM
12  updated by JMH on June 8, 2022 at 8:48 AM
13 -->
14
15 <script>
16   // this function returns a random integer between 0 and max-1, inclusive
17   //   reference: https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/
18   //             Global_Objects/Math/random
19   var getRandomInt = function( max ) {
20     return Math.floor( Math.random() * max ) ;
21   }
22
23   //////////////////////////////////////
24   //// (1) create the original array
25
26   // declare the original array
27   var arrInts = new Array() ;
28
29   // generate random integers between 0 and 99 and populate the original array
30   for ( var k = 0 ; k < 20 ; k++ ) {
31     arrInts.push( getRandomInt( 100 ) ) ;
32   }
33
34   // declare a second array to hold the sorted data
35   var arrAlphaSorted = new Array() ;
36
37   //////////////////////////////////////
38   //// (2) create a duplicate of the original array
39
40   // copy the original array to the second array
41   for ( var k = 0 ; k < arrInts.length ; k++ ) {
42     arrAlphaSorted.push( arrInts[k] ) ;
43   }
44   // shortcut for the above for Loop: var arrAlphaSorted = [...arrInts] ;
45   //   reference: https://www.freecodecamp.org/news/
46   //             how-to-clone-an-array-in-javascript-1d3183468f6a/
47
48   // sort the second array alphabetically
49   arrAlphaSorted.sort() ;
50
51   //////////////////////////////////////
52   //// (3) create another duplicate of the original array
53
54   // copy the original array to a third array using the shortcut
55   var arrNumSorted = [...arrInts] ;
56   // define a comparison function to return:
57   //   a negative number if a should be before b
58   //   a positive number if b should be before a
59   //   0 if there should be no changes to the sort order
60   var comp = function( a, b ) {
61     return a - b ;
62   }
63   // sort the third array numerically
64   arrNumSorted.sort( comp ) ;
65
66 </script>
67

```

```

68 <style>
69   th {
70     width: 8em ;           /* to make all the columns the same width */
71     color: white ;        /* to make all the columns the same with white */
72     background-color: black ; /* Letters on a black background for emphasis */
73     font-weight: bold ;
74   }
75   table tr:nth-child(2) td { /* to center the text in the second row of the table */
76     text-align: center ;
77   }
78   ol {                     /* to position the list of array elements in the table */
79     padding-left: 4em ;
80   }
81   li {                     /* to leave a bit more space between a list item's */
82     padding-left: 0.4em ; /* number and its corresponding element value */
83   }
84   @media print {
85     table {
86       width: 60% ;        /* to make the table a reasonable size when printed and */
87       margin-top: 3em ;   /* give it a bit of space at the top of the page */
88     }
89   }
90 </style>
91
92 </head>
93
94 <body>
95   <table cellpadding="0" cellspacing="5" border="1" align="center">
96     <tr>
97       <!-- display the column heads -->
98       <th>Original<br>Array</th>
99       <th>Sorted<br>Alphabetically</th>
100      <th>Sorted<br>Numerically</th>
101    </tr>
102    <tr>
103      <!-- explain how each list was generated -->
104      <td>generated using the <code>Math.random()</code> function</td>
105      <td>sorted by simply calling the <code>.sort()</code> function</td>
106      <td>sorted by calling <code>.sort(cmp)</code>, where <code>cmp</code> is a comparison function</td>
107    </tr>
108    <tr>
109
110      <!-- display the original array -->
111      <td>
112        <ol start="0">
113          <script>
114            for ( var k = 0 ; k < arrInts.length ; k++ ) {
115              document.writeln( "<li>" + arrInts[k] + "</li>" ) ;
116            }
117          </script>
118        </ol>
119      </td>
120
121      <!-- display the alphabetically sorted array -->
122      <td>
123        <ol start="0">
124          <script>
125            for ( var k = 0 ; k < arrAlphaSorted.length ; k++ ) {
126              document.writeln( "<li>" + arrAlphaSorted[k] + "</li>" ) ;
127            }
128          </script>
129        </ol>
130      </td>
131

```

```
132     <!-- display the numerically sorted array -->
133     <td>
134         <ol start="0">
135             <script>
136                 for ( var k = 0 ; k < arrNumSorted.length ; k++ ) {
137                     document.writeln( "<li>" + arrNumSorted[k] + "</li>" ) ;
138                 }
139             </script>
140         </ol>
141     </td>
142
143 </tr>
144 </table>
145
146 </body>
147
148 </html>
```