

# Array Example 2

Version 3.3, updated February 24, 2023

## This is Code Option # 1

1. Tuesday, January 3, 2023 : HTML Markup for Basic Text
2. Thursday, January 5, 2023 : HTML Structure and More Basic Markup
3. Tuesday, January 10, 2023 : Creating Lists
4. Thursday, January 12, 2023 : Creating Nested and Definition Lists
5. Tuesday, January 17, 2023 : Creating Tables
6. Thursday, January 19, 2023 : Working with Images
7. Tuesday, January 24, 2023 : CSS Structure and Basic Rules
8. Thursday, January 26, 2023 : CSS Structure and Basic Rules (*repeat*)
9. Tuesday, January 31, 2023 : CSS Selectors and the `span` and `div` Elements
10. Thursday, February 2, 2023 : Advanced Text Formatting Styles
11. Tuesday, February 7, 2023 : CSS Inheritance and Priority System
12. Thursday, February 9, 2023 : Individual Assistance Day
13. Tuesday, February 14, 2023 : Using Hyperlinks
14. Thursday, February 16, 2023 : Introduction to JavaScript
15. Tuesday, February 21, 2023 : JavaScript Control Structures
16. Thursday, February 23, 2023 : JavaScript Control Structures (*continued*)
17. Tuesday, February 28, 2023 : Structuring Data with Arrays
18. Thursday, March 2, 2023 : Structuring and Sorting Data Stored in Arrays
19. Tuesday, March 7, 2023 : Working with Two-Dimensional (2D) Arrays
20. Thursday, March 9, 2023 : Writing Your Own JavaScript Functions
21. Tuesday, March 14, 2023 : Introduction to HTML Forms
22. Thursday, March 16, 2023 : No Class Meeting – Instructor Unavailable

# Array Example 2

Version 3.3, updated February 24, 2023

## This is Code Option # 2

1. Tuesday, January 3, 2023 : **HTML Markup for Basic Text**
2. Thursday, January 5, 2023 : **HTML Structure and More Basic Markup**
3. Tuesday, January 10, 2023 : **Creating Lists**
4. Thursday, January 12, 2023 : **Creating Nested and Definition Lists**
5. Tuesday, January 17, 2023 : **Creating Tables**
6. Thursday, January 19, 2023 : **Working with Images**
7. Tuesday, January 24, 2023 : **CSS Structure and Basic Rules**
8. Thursday, January 26, 2023 : **CSS Structure and Basic Rules** (*repeat*)
9. Tuesday, January 31, 2023 : **CSS Selectors and the span and div Elements**
10. Thursday, February 2, 2023 : **Advanced Text Formatting Styles**
11. Tuesday, February 7, 2023 : **CSS Inheritance and Priority System**
12. Thursday, February 9, 2023 : **Individual Assistance Day**
13. Tuesday, February 14, 2023 : **Using Hyperlinks**
14. Thursday, February 16, 2023 : **Introduction to JavaScript**
15. Tuesday, February 21, 2023 : **JavaScript Control Structures**
16. Thursday, February 23, 2023 : **JavaScript Control Structures** (*continued*)
17. Tuesday, February 28, 2023 : **Structuring Data with Arrays**
18. Thursday, March 2, 2023 : **Structuring and Sorting Data Stored in Arrays**
19. Tuesday, March 7, 2023 : **Working with Two-Dimensional (2D) Arrays**
20. Thursday, March 9, 2023 : **Writing Your Own JavaScript Functions**
21. Tuesday, March 14, 2023 : **Introduction to HTML Forms**
22. Thursday, March 16, 2023 : **No Class Meeting – Instructor Unavailable**

```

Line wrap 
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <meta charset="utf-8">
5   <title>Array Example 2</title>
6
7
8 <!--
9   File: cssdweb.edu/StudentResources/Website Development I/CodeSamples/
10  ArrayExample_2_v3_2023-02_option2.html
11  Copyright (c) 2022-2023 by Jesse Heines. All rights reserved. May be freely
12  copied or excerpted for educational purposes with credit to the author.
13  updated by JMH on May 11, 2022 at 8:38 PM
14  updated by JMH on May 31, 2022 at 4:31 PM
15  updated by JMH on June 7, 2022 at 8:36 AM
16  updated by JMH on September 19, 2022 at 1:18 PM
17  updated by JMH on November 27, 2022 at 4:17 PM
18  updated by JMH on February 24, 2023 at 8:52 PM
19 -->
20
21 <script>
22   // month names
23   var arrMonth = // names of all the months (0-based)
24     [ "January", "February", "March", "April", "May", "June",
25       "July", "August", "September", "October", "November", "December" ] ;
26
27   // day names
28   var arrWeekday = // names of all the days of the week (0-based)
29     [ "Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday" ] ;
30
31   var arrClassTopics = [
32     /* # 0 : Tues 01-03 */ "HTML Markup for Basic Text",
33     /* # 1 : Thur 01-05 */ "HTML Structure and More Basic Markup",
34     /* # 2 : Tues 01-10 */ "Creating Lists",
35     /* # 3 : Thur 01-12 */ "Creating Nested and Definition Lists",
36     /* # 4 : Tues 01-17 */ "Creating Tables",
37     /* # 5 : Thur 01-19 */ "Working with Images",
38     /* # 6 : Tues 01-24 */ "CSS Structure and Basic Rules",
39     /* # 7 : Thur 01-26 */ "CSS Structure and Basic Rules <span style='font-weight: normal'><em>repeat</em></span>",
40     /* # 8 : Tues 01-31 */ "CSS Selectors and the <code>span</code> and <code>div</code> Elements",
41     /* # 9 : Thur 02-02 */ "Advanced Text Formatting Styles",
42     /* #10 : Tues 02-07 */ "CSS Inheritance and Priority System",
43     /* #11 : Thur 02-09 */ "Individual Assistance Day",
44     /* #12 : Tues 02-14 */ "Using Hyperlinks",
45     /* #13 : Thur 02-16 */ "Introduction to JavaScript",
46     /* #14 : Tues 02-21 */ "JavaScript Control Structures",
47     /* #15 : Thur 02-23 */ "JavaScript Control Structures (<em>continued</em>)",
48     /* #16 : Tues 02-28 */ "Structuring Data with Arrays",
49     /* #17 : Thur 03-02 */ "Structuring and Sorting Data <span class='nowrap'>Stored in Arrays</span>",
50     /* #18 : Tues 03-07 */ "Working with Two-Dimensional (2D) Arrays",
51     /* #19 : Thur 03-09 */ "Writing Your Own JavaScript Functions",
52     /* #20 : Tues 03-14 */ "Introduction to HTML Forms",
53     /* #21 : Thur 03-16 */ "No Class Meeting &ndash; Instructor Unavailable",
54   ] ;
55 </script>
56
57 <style>
58   #tblSchedule tr td {
59     text-align : right ;
60   }
61   #tblSchedule tr td:last-child {
62     text-align : left ;
63   }
64 </style>
65 </head>
66
67 <body>
68   <h2 style="margin-bottom: 0.2em">Array Example 2</h2>
69   <p style="margin-top: 0; margin-bottom: 1.5em">Version 3.3, updated February 24, 2023</p>
70   <script>
71     // specify which code option you want to execute
72     var nCodeOption = 2 ;
73     document.writeln( "<h3>This is Code Option # " + nCodeOption + "</h3>" ) ;
74   </script>
75   <p></p>
76
77
78
79
80

```

```

81 <script>
82 // The following is DEBUGGING CODE. The Boolean expression as written is false, so
83 // the code that it controls is not executed. This technique is used to keep debugging
84 // code in a program (instead of deleting it) but to prevent it from being executed.
85 if ( false ) {
86 // remember that the month is 0-based, thus, January is month #0
87 var dateFirstClass = new Date( 2023, 0, 3 );
88 console.log( dateFirstClass );
89 // appending a Date object to a blank string ("" ) forces the result to be a
90 // string, from which we can then extract information with the built-in
91 // substr[ing] function
92 // this is possible because the date string has a standard format
93 console.log( (""+dateFirstClass).substr( 4, 3 ) ); // 3-character month
94 console.log( (""+dateFirstClass).substr( 8, 2 ) ); // 2-character day
95 console.log( dateFirstClass.getDate() ); // day of the month
96 console.log( dateFirstClass.getFullYear() ); // 4-character year
97 }
98 </script>
99
100 <table id="tblSchedule" cellpadding="0" border="0">
101 <script>
102 // remember that the month is 0-based, thus, January is month #0
103 var dateFirstClass = new Date( 2023, 0, 3 );
104 var date = dateFirstClass ;
105
106 // loop from 0 to one less than the length of the array
107 for ( var k = 0 ; k < arrClassTopics.length ; k++ ) {
108
109 document.writeln( "<tr>" ) ; // start a new table row
110
111 switch ( nCodeOption ) {
112
113 case 1 : // code option 1, testing all the data and arrays
114 document.writeln( " <td>" + (k+1) + ". " + arrWeekday[date.getDay()] + ", " +
115 arrMonth[date.getMonth()] + " " + date.getDate() + ", " +
116 date.getFullYear() + " : " + arrClassTopics[k] + "</td>" ) ;
117 break ;
118
119 case 2 : // code option 2, to allow for CSS control of each element
120 document.writeln( " <td>" + (k+1) + ".&nbsp;</td>" ) ;
121 document.writeln( " <td>" + arrWeekday[date.getDay()] + "&nbsp;</td>" ) ;
122 document.writeln( " <td>" + arrMonth[date.getMonth()] + "&nbsp;</td>" ) ;
123 document.writeln( " <td>" + date.getDate() + "&nbsp;</td>" ) ;
124 document.writeln( " <td>" + date.getFullYear() + " : " ) ;
125 if ( arrClassTopics[k] == "No Class Meeting" ) {
126 document.writeln( "<em style='color: red'>" ) ;
127 } else {
128 document.writeln( "<strong>" ) ;
129 }
130 document.writeln( arrClassTopics[k] ) ;
131 if ( arrClassTopics[k] == "No Class Meeting" ) {
132 document.writeln( "</em>" ) ;
133 } else {
134 document.writeln( "</strong>" ) ;
135 }
136 document.writeln( "</td>" ) ;
137 break ;
138
139 } // end switch
140
141 document.writeln( "</tr>" ) ; // end the current table row
142
143 // set the date to the date of the next class
144 // reference: https://stackoverflow.com/questions/563406/how-to-add-days-to-date
145 // The modulus (%) function controls whether we add 2 days (from Tuesday to
146 // Thursday) or 5 days (from Thursday to the Tuesday of the next week).
147 if ( k % 2 == 0 ) {
148 date.setDate( date.getDate() + 2 ) ;
149 } else {
150 date.setDate( date.getDate() + 5 ) ;
151 }
152 }
153 </script>
154 </table>
155 </body>
156 </html>
157
158

```