

Array Example 2

Version 3, updated September 19, 2022

This is Code Option # 1

1. Tuesday, July 5, 2022 : No Class Meeting
2. Thursday, July 7, 2022 : No Class Meeting
3. Tuesday, July 12, 2022 : HTML Markup for Basic Text
4. Thursday, July 14, 2022 : No Class Meeting
5. Tuesday, July 19, 2022 : HTML Structure & More Basic Markup
6. Thursday, July 21, 2022 : No Class Meeting
7. Tuesday, July 26, 2022 : No Class Meeting
8. Thursday, July 28, 2022 : Creating Lists & Using Entities
9. Tuesday, August 2, 2022 : Working with Images
10. Thursday, August 4, 2022 : CSS Structure & Basic Rules
11. Tuesday, August 9, 2022 : No Class Meeting
12. Thursday, August 11, 2022 : No Class Meeting
13. Tuesday, August 16, 2022 : Following Good Coding Practices, Review of CSS Structure, & CSS Selectors
14. Thursday, August 18, 2022 : CSS Units, Borders, & Backgrounds
15. Tuesday, August 23, 2022 : No Class Meeting
16. Thursday, August 25, 2022 : No Class Meeting
17. Tuesday, August 30, 2022 : No Class Meeting
18. Thursday, September 1, 2022 : No Class Meeting
19. Tuesday, September 6, 2022 : Working with Tables
20. Thursday, September 8, 2022 : Using Hyperlinks
21. Tuesday, September 13, 2022 : CSS Selector Techniques & Priority System
22. Thursday, September 15, 2022 : Advanced Text Formatting Styles
23. Tuesday, September 20, 2022 : Introduction to JavaScript
24. Thursday, September 22, 2022 : Built-In JavaScript Functions
25. Tuesday, September 27, 2022 : Writing Your Own Functions
26. Thursday, September 29, 2022 : Individual Work to Bring Projects to Closure

Array Example 2

Version 3, updated September 19, 2022

This is Code Option # 2

1. Tuesday, July 5, 2022 : *No Class Meeting*
2. Thursday, July 7, 2022 : *No Class Meeting*
3. Tuesday, July 12, 2022 : **HTML Markup for Basic Text**
4. Thursday, July 14, 2022 : *No Class Meeting*
5. Tuesday, July 19, 2022 : **HTML Structure & More Basic Markup**
6. Thursday, July 21, 2022 : *No Class Meeting*
7. Tuesday, July 26, 2022 : *No Class Meeting*
8. Thursday, July 28, 2022 : **Creating Lists & Using Entities**
9. Tuesday, August 2, 2022 : **Working with Images**
10. Thursday, August 4, 2022 : **CSS Structure & Basic Rules**
11. Tuesday, August 9, 2022 : *No Class Meeting*
12. Thursday, August 11, 2022 : *No Class Meeting*
13. Tuesday, August 16, 2022 : **Following Good Coding Practices, Review of CSS Structure, & CSS Selectors**
14. Thursday, August 18, 2022 : **CSS Units, Borders, & Backgrounds**
15. Tuesday, August 23, 2022 : *No Class Meeting*
16. Thursday, August 25, 2022 : *No Class Meeting*
17. Tuesday, August 30, 2022 : *No Class Meeting*
18. Thursday, September 1, 2022 : *No Class Meeting*
19. Tuesday, September 6, 2022 : **Working with Tables**
20. Thursday, September 8, 2022 : **Using Hyperlinks**
21. Tuesday, September 13, 2022 : **CSS Selector Techniques & Priority System**
22. Thursday, September 15, 2022 : **Advanced Text Formatting Styles**
23. Tuesday, September 20, 2022 : **Introduction to JavaScript**
24. Thursday, September 22, 2022 : **Built-In JavaScript Functions**
25. Tuesday, September 27, 2022 : **Writing Your Own Functions**
26. Thursday, September 29, 2022 : **Individual Work to Bring Projects to Closure**

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4   <meta charset="utf-8">
5   <title>Array Example 2</title>
6
7 <!--
8   File:  cssdweb.edu/StudentResources/Website Development I/CodeSamples/ArrayExample_2.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 May 11, 2022 at 8:38 PM
12  updated by JMH on May 31, 2022 at 4:31 PM
13  updated by JMH on June 7, 2022 at 8:36 AM
14  updated by JMH on September 19, 2022 at 1:18 PM
15 -->
16
17 <script>
18   // month names
19   var arrMonth = // names of all the months (0-based)
20     [ "January", "February", "March", "April", "May", "June",
21       "July", "August", "September", "October", "November", "December" ] ;
22
23   // day names
24   var arrWeekday = // names of all the days of the week (0-based)
25     [ "Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday" ] ;
26
27   var arrClassTopics = [
28     /* 0 */ "No Class Meeting",
29     /* 1 */ "No Class Meeting",
30     /* 2 */ "HTML Markup for Basic Text",
31     /* 3 */ "No Class Meeting",
32     /* 4 */ "HTML Structure & More Basic Markup",
33     /* 5 */ "No Class Meeting",
34     /* 6 */ "No Class Meeting",
35     /* 7 */ "Creating Lists & Using Entities",
36     /* 8 */ "Working with Images",
37     /* 9 */ "CSS Structure & Basic Rules",
38     /* 10 */ "No Class Meeting",
39     /* 11 */ "No Class Meeting",
40     /* 12 */ "Following Good Coding Practices, Review of CSS Structure, & CSS Selectors",
41     /* 13 */ "CSS Units, Borders, & Backgrounds",
42     /* 14 */ "No Class Meeting",
43     /* 15 */ "No Class Meeting",
44     /* 16 */ "No Class Meeting",
45     /* 17 */ "No Class Meeting",
46     /* 18 */ "Working with Tables",
47     /* 19 */ "Using Hyperlinks",
48     /* 20 */ "CSS Selector Techniques & Priority System",
49     /* 21 */ "Advanced Text Formatting Styles",
50     /* 22 */ "Introduction to JavaScript",
51     /* 23 */ "Built-In JavaScript Functions",
52     /* 24 */ "Writing Your Own Functions",
53     /* 25 */ "Individual Work to Bring Projects to Closure",
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, updated September 19, 2022</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 <script>
79     // The following is DEBUGGING CODE. The Boolean expression as written is false, so
80     // the code that it controls is not executed. This technique is used to keep debugging
81     // code in a program (instead of deleting it) but to prevent it from being executed.
82     if ( false ) {
83         // remember that the month is 0-based, thus, July is month #6
84         var dateFirstClass = new Date( 2022, 6, 5 ) ;
85         console.log( dateFirstClass ) ;
86         // appending a Date object to a blank string ("") forces the result to be a
87         // string, from which we can then extract information with the built-in
88         // substr[ing] function
89         // this is possible because the date string has a standard format
90         console.log( (""+dateFirstClass).substr( 4, 3 ) ) ; // 3-character month
91         console.log( (""+dateFirstClass).substr( 8, 2 ) ) ; // 2-character day
92         console.log( dateFirstClass.getDate() ) ;           // day of the month
93         console.log( dateFirstClass.getFullYear() ) ;       // 4-character year
94     }
95 </script>
96
97 <table id="tblSchedule" cellspacing="0" border="0">
98     <script>
99         // remember that the month is 0-based, thus, July is month #6
100        var dateFirstClass = new Date( 2022, 6, 5 ) ;
101        var date = dateFirstClass ;
102
103        // Loop from 1 to the length of the array
104        for ( var k = 0 ; k < arrClassTopics.length ; k++ ) {
105            document.writeln( "<tr>" ) ;
106
107            switch ( nCodeOption ) {
108
109                case 1 : // code option 1, testing all the data and arrays
110                    document.writeln( " <td>" + (k+1) + ". " + arrWeekday[date.getDay()] + ", " +
111                        arrMonth[date.getMonth()] + " " + date.getDate() + ", " +
112                        date.getFullYear() + " : " + arrClassTopics[k] + "</td>" ) ;
113                    break ;
114
115                case 2 : // code option 2, to allow for CSS control of each element
116                    document.writeln( " <td>" + (k+1) + ".&nbsp;</td>" ) ;
117                    document.writeln( " <td>" + arrWeekday[date.getDay()] + ",&nbsp;</td>" ) ;
118                    document.writeln( " <td>" + arrMonth[date.getMonth()] + "&nbsp;</td>" ) ;
119                    document.writeln( " <td>" + date.getDate() + ",&nbsp;</td>" ) ;
120                    document.writeln( " <td>" + date.getFullYear() + " : " ) ;
121                    if ( arrClassTopics[k] == "No Class Meeting" ) {
122                        document.writeln( "<em style='color: red;'>" ) ;
123                    } else {
124                        document.writeln( "<strong>" ) ;
125                    }
126                    document.writeln( arrClassTopics[k] ) ;
127                    if ( arrClassTopics[k] == "No Class Meeting" ) {
128                        document.writeln( "</em>" ) ;
129                    } else {

```

```
130         document.writeln( "</strong>" ) ;
131     }
132     document.writeln( "</td>" ) ;
133     break ;
134 }
135
136 document.writeln( "</tr>" ) ;
137
138     // set the date to the date of the next class
139     // reference: https://stackoverflow.com/questions/563406/how-to-add-days-to-date
140     // The modulus (%) function controls whether we add 2 days (from Tuesday to
141     // Thursday) or 5 days (from Thursday to the Tuesday of the next week).
142     if ( k % 2 == 0 ) {
143         date.setDate( date.getDate() + 2 ) ;
144     } else {
145         date.setDate( date.getDate() + 5 ) ;
146     }
147 }
148 </script>
149 </table>
150 </body>
151
152 </html>
```