

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <meta charset="utf-8">
5   <title>Array Example 2</title>
6
7 <!--
8   File:  cssdweb.edu/Student Resources/Website Development I/CodeSamples/ArrayExample_2.html
9   maintained by Jesse M. Heines, jesse@jesseheines.com
10  Copyright (c) 2022 by Jesse Heines. All rights reserved. May be freely
11     copied or excerpted for educational purposes with credit to the author.
12  updated by JMH on May 11, 2022 at 8:38 PM
13 -->
14
15 <script>
16   // month names
17   var arrMonth = // names of all the months (0-based)
18     [ "January", "February", "March", "April", "May", "June",
19       "July", "August", "September", "October", "November", "December" ] ;
20
21   // day names
22   var arrWeekday = // names of all the days of the week (0-based)
23     [ "Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday" ] ;
24
25   var arrClassTopics = [
26     /* 0 */ "Markup for Basic Text",
27     /* 1 */ "HTML Structure & More Basic Markup " ,
28     /* 2 */ "Creating Lists & Using Entities",
29     /* 3 */ "CSS Structure & Basic Rules",
30     /* 4 */ "Working with Images",
31     /* 5 */ "No Class Meeting",
32     /* 6 */ "CSS Selectors, Units, Borders & Backgrounds",
33     /* 7 */ "No Class Meeting",
34     /* 8 */ "Working with Tables",
35     /* 9 */ "Using Hyperlinks",
36     /* 10 */ "CSS Selector Techniques & Priority System",
37     /* 11 */ "No Class Meeting",
38     /* 12 */ "Advanced Text Formatting Styles",
39     /* 13 */ "Introduction to JavaScript",
40     /* 14 */ "Built-In JavaScript Functions",
41     /* 15 */ "Writing Your Own Functions",
42     /* 16 */ "Manipulating Page Objects",
43     /* 17 */ "Structuring JavaScript Data",
44     /* 18 */ "Introduction to jQuery",
45     /* 19 */ "Working with Dynamic Objects",
46     /* 20 */ "Introduction to Forms",
47     /* 21 */ "Retrieving and Processing Form Responses",
48     /* 22 */ "Validating Form Responses and Providing Feedback",
49     /* 23 */ "Project Presentations"
50   ] ;
51 </script>
52
53 <style>
54   #tblSchedule tr td {
55     text-align : right ;
56   }
57   #tblSchedule tr td:last-child {
58     text-align : left ;
59   }
60 </style>
61 </head>
62
```

```

63 <body>
64   <script>
65     // The following is debugging code. The Boolean expression as written is false, so
66     // the code that it controls is not executed. This technique is used to keep debugging
67     // code in a program (instead of deleting it) but to prevent it from being executed.
68     if ( 0 == 1 ) {
69       // remember that the month is 0-based, thus, April is month #3
70       var dateFirstClass = new Date( 2022, 3, 5 );
71       console.log( dateFirstClass );
72       // appending a Date object to a blank string ("" ) forces the result to be a
73       // string, from which we can then extract information with the built-in
74       // substr[ing] function
75       // this is possible because the date string has a standard format
76       console.log( (""+dateFirstClass).substr( 4, 3 ) ); // 3-character month
77       console.log( (""+dateFirstClass).substr( 8, 2 ) ); // 2-character day
78       console.log( dateFirstClass.getDate() ); // day of the month
79       console.log( dateFirstClass.getFullYear() ); // 4-character year
80     }
81   </script>
82
83   <table id="tblSchedule" cellpadding="0" border="0">
84     <script>
85       // remember that the month is 0-based, thus, April is month #3
86       var dateFirstClass = new Date( 2022, 3, 5 );
87       var date = dateFirstClass ;
88
89       // specify which version you want to execute
90       var version = 2 ;
91
92       // Loop from 1 to 24
93       for ( var k = 1 ; k < 25 ; k++ ) {
94         document.writeln( "<tr>" );
95
96         switch ( version ) {
97
98           case 1 : // version 1, testing all the data and arrays
99             document.writeln( " <td>" + k + ". " + arrWeekday[date.getDay()] + ", " +
100               arrMonth[date.getMonth()] + " " + date.getDate() + ", " +
101               date.getFullYear() + " : " + arrClassTopics[k-1] + "</td>" );
102             break ;
103
104           case 2 : // version 2, to allow for CSS control of each element
105             document.writeln( " <td>" + k + ".&nbsp;</td>" );
106             document.writeln( " <td>" + arrWeekday[date.getDay()] + "&nbsp;</td>" );
107             document.writeln( " <td>" + arrMonth[date.getMonth()] + "&nbsp;</td>" );
108             document.writeln( " <td>" + date.getDate() + "&nbsp;</td>" );
109             document.writeln( " <td>" + date.getFullYear() + " : " );
110             if ( arrClassTopics[k-1] == "No Class Meeting" ) {
111               document.writeln( "<em style='color: red;'>" );
112             } else {
113               document.writeln( "<strong>" );
114             }
115             document.writeln( arrClassTopics[k-1] );
116             if ( arrClassTopics[k-1] == "No Class Meeting" ) {
117               document.writeln( "</em>" );
118             } else {
119               document.writeln( "</strong>" );
120             }
121             document.writeln( "</td>" );
122             break ;
123         }
124
125         document.writeln( "</tr>" );

```

```
126
127 // set the date to the date of the next class
128 // reference: https://stackoverflow.com/questions/563406/how-to-add-days-to-date
129 // The modulus (%) function controls whether we add 2 days (from Tuesday to
130 // Thursday) or 5 days (from Thursday to the Tuesday of the next week).
131 if ( k % 2 == 1 ) {
132     date.setDate( date.getDate() + 2 ) ;
133 } else {
134     date.setDate( date.getDate() + 5 ) ;
135 }
136 }
137 </script>
138 </table>
139 </body>
140
141 </html>
```

1. Tuesday, April 5, 2022 : Markup for Basic Text
2. Thursday, April 7, 2022 : HTML Structure & More Basic Markup
3. Tuesday, April 12, 2022 : Creating Lists & Using Entities
4. Thursday, April 14, 2022 : CSS Structure & Basic Rules
5. Tuesday, April 19, 2022 : Working with Images
6. Thursday, April 21, 2022 : No Class Meeting
7. Tuesday, April 26, 2022 : CSS Selectors, Units, Borders & Backgrounds
8. Thursday, April 28, 2022 : No Class Meeting
9. Tuesday, May 3, 2022 : Working with Tables
10. Thursday, May 5, 2022 : Using Hyperlinks
11. Tuesday, May 10, 2022 : CSS Selector Techniques & Priority System
12. Thursday, May 12, 2022 : No Class Meeting
13. Tuesday, May 17, 2022 : Advanced Text Formatting Styles
14. Thursday, May 19, 2022 : Introduction to JavaScript
15. Tuesday, May 24, 2022 : No Class Meeting
16. Thursday, May 26, 2022 : Individual Work
17. Tuesday, May 31, 2022 : Built-In JavaScript Functions
18. Thursday, June 2, 2022 : Writing Your Own Functions
19. Tuesday, June 7, 2022 : Manipulating Page Objects
20. Thursday, June 9, 2022 : Structuring JavaScript Data
21. Tuesday, June 14, 2022 : Introduction to jQuery
22. Thursday, June 16, 2022 : Working with Dynamic Objects
23. Tuesday, June 21, 2022 : Introduction to Forms
24. Thursday, June 23, 2022 : Retrieving and Processing Form Responses

1. Tuesday, April 5, 2022 : **Markup for Basic Text**
2. Thursday, April 7, 2022 : **HTML Structure & More Basic Markup**
3. Tuesday, April 12, 2022 : **Creating Lists & Using Entities**
4. Thursday, April 14, 2022 : **CSS Structure & Basic Rules**
5. Tuesday, April 19, 2022 : **Working with Images**
6. Thursday, April 21, 2022 : *No Class Meeting*
7. Tuesday, April 26, 2022 : **CSS Selectors, Units, Borders & Backgrounds**
8. Thursday, April 28, 2022 : *No Class Meeting*
9. Tuesday, May 3, 2022 : **Working with Tables**
10. Thursday, May 5, 2022 : **Using Hyperlinks**
11. Tuesday, May 10, 2022 : **CSS Selector Techniques & Priority System**
12. Thursday, May 12, 2022 : *No Class Meeting*
13. Tuesday, May 17, 2022 : **Advanced Text Formatting Styles**
14. Thursday, May 19, 2022 : **Introduction to JavaScript**
15. Tuesday, May 24, 2022 : *No Class Meeting*
16. Thursday, May 26, 2022 : **Individual Work**
17. Tuesday, May 31, 2022 : **Built-In JavaScript Functions**
18. Thursday, June 2, 2022 : **Writing Your Own Functions**
19. Tuesday, June 7, 2022 : **Manipulating Page Objects**
20. Thursday, June 9, 2022 : **Structuring JavaScript Data**
21. Tuesday, June 14, 2022 : **Introduction to jQuery**
22. Thursday, June 16, 2022 : **Working with Dynamic Objects**
23. Tuesday, June 21, 2022 : **Introduction to Forms**
24. Thursday, June 23, 2022 : **Retrieving and Processing Form Responses**