

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="utf-8">
5   <!--
6     File: ~/heines/91.461/resources/Book-MacDonald-HTML5/Chapter04/ProgressBarWithTimer.html
7     Jesse M. Heines, UMass Lowell Computer Science, heines@cs.uml.edu
8     Copyright (c) 2012 by Jesse M. Heines. All rights reserved. May be freely
9     copied or excerpted for educational purposes with credit to the author.
10    updated by JMH on October 31, 2012 at 9:40 AM
11    updated by JMH on November 11, 2012 at 8:57 PM
12    updated by JMH on November 12, 2012 at 2:14 PM
13  -->
14  <title>Progress Bar with Button and Timer</title>
15  <style>
16    body {
17      font-family: 'Palatino Linotype', serif;
18    }
19  </style>
20
21  <script type="text/javascript" src="/~heines/jquery/jquery-1.8.1.min.js"></script>
22  <script type="text/javascript">
23    $(document).ready( function() {
24      $( "#minus10" ).attr( { "disabled" : true } );
25      $( "#msg1" ).html( "<em>The current progress value is:</em>&nbsp; " +
26        $( "#progress1" ).val() );
27      $( "#msg2" ).html( "<em>The timer is ready to run." );
28
29      // Note that the disabled property is only available in code.
30      // If you use attribute disabled="false" in HTML, there is no effect, since
31      // the mere presence of the disabled attribute disables a control, making
32      // disabled="false" exactly the same as disabled="true".
33      // Also, the disabled property stays set even when you reload the page, at
34      // least in Firefox.
35      // Thus, the Start button must be enabled in code when the page is first loaded
36      // to ensure that it is available.
37      // Finally, note that the value to which the disabled attribute is set must be
38      // of type boolean, either true or false. The string values "true" and "false"
39      // will not work as expected: they will both be interpreted as false and
40      // disable the button.
41      $( "#start" ).attr( { "disabled" : false } );
42      $( "#reset" ).css( { "visibility" : "hidden" } );
43
44      // This function increments the 1st progress bar by 10 when the "Increase
45      // Progress by 10%" button is clicked.
46      // Note that there is no need to check for the progress bar value exceeding 100.
47      // This check appears to be built in, at least in Firefox, because clicking
48      // the "Increase Progress by 10%" button has no effect when the progress bar's
49      // value is already 100.
50      $( "#plus10" ).click( function() {
51        $( "#progress1" ).val( $( "#progress1" ).val() + 10 );
52        $( "#msg1" ).html( "<em>The current progress value is:</em>&nbsp; " +
53          $( "#progress1" ).val() );
54        if ( $( "#progress1" ).val() >= 100 ) {
55          $(this).attr( { "disabled" : true } );
56        } else if ( $( "#progress1" ).val() >= 0 ) {
57          $( "#minus10" ).attr( { "disabled" : false } );
58        } ;
59      } );
60    } );
61
62    // This function decrements the 1st progress bar by 10 when the "Decrease
63    // Progress by 10%" button is clicked.
64    // Note that there is no need to check for the progress bar value being 0 or less.
65    // This check appears to be built in, at least in Firefox, because clicking
66    // the "Decrease Progress by 10%" button has no effect when the progress bar's
67    // value is already 0.
```

```

68     $( "#minus10" ).click( function() {
69         $( "#progress1" ).val( $( "#progress1" ).val() - 10 );
70         $( "#msg1" ).html( "<em>The current progress value is:</em>&nbsp;   " +
71             $( "#progress1" ).val() );
72         if ( $( "#progress1" ).val() <= 0 ) {
73             $(this).attr( { "disabled" : true } );
74         } else if ( $( "#progress1" ).val() < 100 ) {
75             $( "#plus10" ).attr( { "disabled" : false } );
76         } ;
77     } );
78
79     // This function increments the 2nd progress bar by 1 each time it is called.
80     // When the progress bar value = 100, the timer triggering this function is
81     // cleared and the Reset button is displayed.
82     function TimerPulse_Plus1() {
83         $( "#progress2" ).val( $( "#progress2" ).val() + 1 );
84         $( "#msg2" ).html( "<em>The timer is currently at " + $( "#progress2" ).val() +
85             "%.</em>" );
86         if ( $( "#progress2" ).val() >= 100 ) {
87             window.clearInterval( myTimer ); // "window." is optional
88             // manually click the Start button to synch jQuery's internal toggle counter
89             $( "#start" ).click();
90             $( "#msg2" ).html( "<em>The timer has reached 100%.</em>" );
91             $( "#start" ).attr( { "disabled" : true } );
92             $( "#reset" ).css( { "visibility" : "visible" } );
93         }
94     }
95
96     // This variable holds a reference to the timer instance.
97     var myTimer ;
98
99     // This function controls the timer that controls the 2nd progress bar.
100    $( "#start" ).toggle(
101        function() {
102            TimerPulse_Plus1();
103            $( "#start" ).val( "Stop Increasing Progress" );
104            myTimer = window.setInterval( TimerPulse_Plus1, 100 ); // "window." is optional
105        },
106        function() {
107            $( "#start" ).val( "Start Increasing Progress" );
108            window.clearInterval( myTimer ); // "window." is optional
109            $( "#msg2" ).html( "<em>The timer has been paused.</em>" );
110        }
111    );
112
113    // This function decrements the 1st progress bar by 10 when the "Decrease
114    // Progress by 10%" button is clicked.
115    $( "#reset" ).click( function() {
116        $( "#progress2" ).val( 0 );
117        $( "#start" ).attr( { "disabled" : false } );
118        $( "#reset" ).css( { "visibility" : "hidden" } );
119        $( "#msg2" ).html( "<em>The timer is ready to run.</em>" );
120    } );
121
122    } );
123 </script>
124 </head>
125

```

