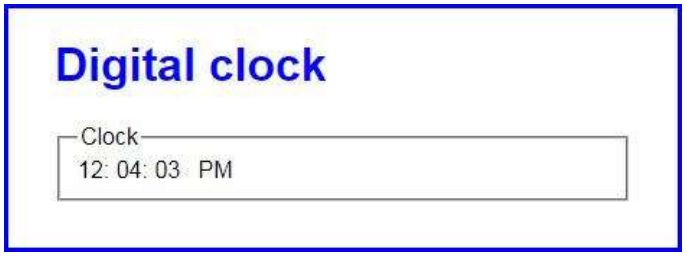
Extra 7-1 Develop the Clock application

In this exercise, you'll create an application that displays the current time in hours, minutes, and seconds. The display should use a 12-hour clock and indicate whether it's AM or PM. The application looks like this:



To convert the computer's time from a 24-hour clock to a 12-hour clock, first check to see if the hours value is greater than 12. If so, subtract 12 from the hours value and set the AM/PM value to "PM". Also, be aware that the hours value for midnight is 0.

1. Open the application in this folder:

   `exercises_extra\ch07\clock\`

2. In the JavaScript file, note that four functions are supplied. The $() function that you can use to select elements. The padSingleDigit() function, that adds a leading zero to single digits using a string method that you'll learn about in chapter 12. The start of a displayCurrentTime() function. And the start of a DOMContentLoaded event handler.

3. In the displayCurrentTime() function, add code that uses the Date object to determine the current hour, minute, and second. Convert these values to a 12-hour clock, determine the AM/PM value, and display these values in the appropriate span tags.

4. In the DOMContentLoaded event handler, code a timer that calls the displayCurrentTime() function at 1 second intervals. Also, make sure that the current time shows as soon as the page loads.

Extra 7-2 Add a stopwatch to the Clock application

In this exercise, you'll add a stopwatch feature to the application you created in extra exercise 7-1. The stopwatch will display elapsed minutes, seconds, and milliseconds. The enhanced application looks like this:

1. Open the application in this folder:

   `exercises_extra\ch07\clock_stopwatch\`

2. In the JavaScript file, note the $(), displayCurrentTime(), padSingleDigit(), and DOMContentLoaded event handler functions from the Clock application. In addition, note the global variables and starting code for the tickStopwatch(), startStopwatch(), stopStopwatch(), and resetStopwatch() functions.
3. In the tickStopwatch() function, add code that adds 10 milliseconds to the elapsedMilliseconds variable and then adjusts the elapsedMinutes and elapsedSeconds variables accordingly. Then, add code that displays the result in the appropriate span tags in the page.
4. In the startStopwatch() function, add code that starts the stopwatch. Be sure to cancel the default action of the link too.
5. In the stopStopwatch() and resetStopwatch() functions, add code that stops the stopwatch. Also, in the resetStopwatch() function, reset the elapsed time and the page display. Be sure to cancel the default action of the links too.
6. In the DOMContentLoaded event handler, attach the stopwatch event handlers to the appropriate links.