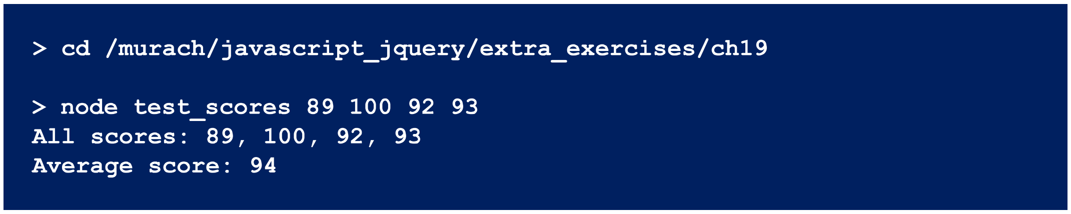
Extra 19-1 Create and run a script with parameters

In this exercise, you'll finish the JavaScript for a server-side script named test_scores that calculates the average of the test scores that are passed to it. You should be able to pass one or more test scores to this script.



```
> cd /murach/javascript_jquery/extra_exercises/ch19

> node test_scores 89 100 92 93
All scores: 89, 100, 92, 93
Average score: 94
```

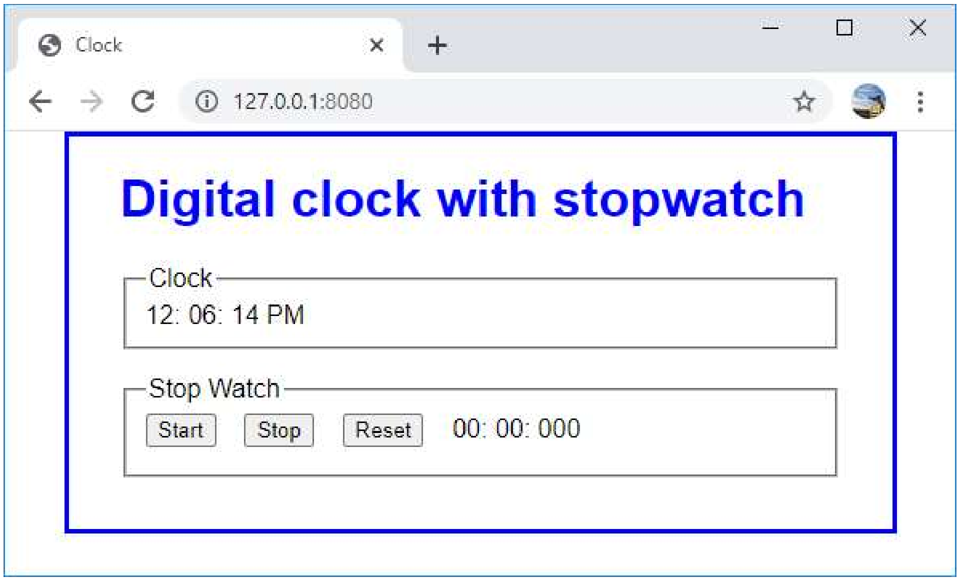
1. If necessary, install Node.js on your system.
2. Open the index.js file in this directory:

   `exercises_extra\ch19\test_scores\`

3. The index.js file contains an empty array named scores, a function named displayScores() that converts the array to a comma-separated string, and a function named calculateAverage() that calculates the average of the scores stored in the scores array.
4. Add code to the JavaScript file that loops through the process.argv array and stores the values entered by the user in the scores array. Remember to skip the first two elements in the process.argv array.
5. Add code that displays all the scores and the average score in the console. Use the displayScores() and calculateAverage() functions to do that.
6. Start the command line for your operating system. Then, use the cd command to change the current directory to the extra_exercises/ch19 directory that contains the test_scores directory.
7. Use the node command to run the index.js script. Test this script to make sure it can handle any number of test score parameters. For example, test it with 2 test scores, and test it with 5 test scores.

Extra 19-2 Run a Clock application that uses ES modules

In this exercise, you'll create a web server with node.js so you can run the Clock application you reviewed in extra exercise 17-3:



Digital clock with stopwatch

Clock

12: 06: 14 PM

Stop Watch

00: 00: 000

1. If necessary, install Node.js on your system.
2. Start a command line for your system.
3. If necessary, use the npm install command to install the http-server module globally.
4. Use a command like the following cd command to change to the ch17/clock_modules directory:

```
cd /murach/javascript_jquery/exercises_extra/ch17/clock_modules
```

5. Use the http-server command to start the web server. If your firewall displays a message asking whether you want to allow Node.js to communicate on private networks, click Allow. If successful, this should display a message that indicates that the server is running. Make a note of the URL in the message.
6. Open a web browser and enter the URL from the last step in the address bar. Be sure to include the port number that comes after the colon.
7. Test the application and make sure the clock and stopwatch work correctly. When you're done, close the browser.
8. Switch to the command line and press Ctrl-C to stop the server. This should display a message that indicates that the server was stopped.