

About Don's Up/Down Timer: What It Is And How To Use It

Documentation for Version 0.99

Donald Byrd

Woodrow Wilson Indiana Teaching Fellow
Adjunct Associate Professor of Informatics, Indiana University Bloomington
donbyrd@indiana.edu

rev. late January 2013

Introduction

I call my Up/Down Timer a “count up/down” timer (as opposed to a “countdown” or “count-down/up” timer, etc.) because it’s designed primarily for use in situations where you want to measure an unpredictable interval of time, then set a timer to go off when the same interval elapses again. I ran into one such situation myself as a student teacher in a middle-school classroom a few years ago where some of my students repeatedly misbehaved and, to discourage them, I wanted to make them do something they wouldn’t like—namely stay after class—for the same amount of time. I looked for a program to help with this, and preferably one written in JavaScript, since then it could run in almost any Web browser with no installation. (Another classroom application might be the opposite scenario, using the timer as a “carrot” rather than a “stick”: The teacher wants to encourage students to do something they don’t like, so the longer they do that, the longer they get to do something they *do* want to do.)

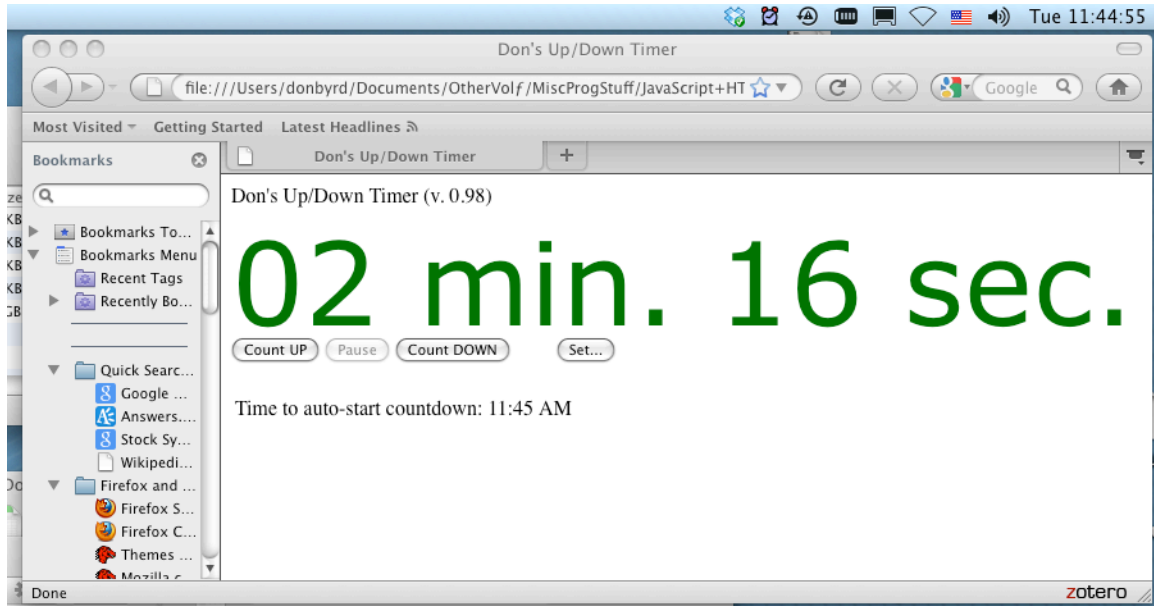
I was certain programs already existed that could do what I wanted, probably even programs in JavaScript; but I couldn’t find one. Every timing program I looked at treated going up and down as unrelated functions, often calling them “stopwatch” and “timer”. (Aside: This looks to me like an example of excessive modality, in the human-computer interaction sense of “mode”.) In any case, Don’s Up/Down Timer is my contribution. Of course, it can also be used as a simple stopwatch (though with less accuracy than most) or timer.

I’ve tested Don’s Up/Down Timer on several Macintosh computers with Firefox and Safari, and on Windows 7 with Firefox and Internet Explorer. It owes a lot, code as well as ideas, to the nice JavaScript Countdown/Count-up Timer/Clock/Ticker for Web Pages by Robert Hashemian (<http://www.hashemian.com/>). Hashemian says of his work: “You can use this code in any manner so long as the author’s name, Web address and this disclaimer is kept intact.”

Using The Timer

The timer functions, written in JavaScript, reside in a file called “DonsUpDownTimer.js”. They must be invoked from HTML code, i.e., a Web page. However, it need not really be a page on the World Wide Web; it can simply be a file on your computer. Two sample files of this sort that are perfectly usable as is are DonsUpDownTimer_Simple.html and DonsUpDownTimer_Fancy.html. As the names suggest, the former uses the timer in a very simple way, not using any of its options, while the latter is much fancier and uses lots of options. One option it uses is “FinishSound”, and that requires an audio file called FinishSound.wav; a simple example, containing one note, comes with the timer files.

To run the timer, just copy DonsUpDownTimer.js, either or both HTML files, and, if needed, FinishSound.wav, into the same directory (folder); then open one of the HTML files with a Web browser. (You can actually run it directly from my Web site, but for several reasons, it's better to make your own copy.)



The Count UP, Pause, and Count DOWN buttons do just what they sound like. The Set button allows changing the “counter” (elapsed time display) to whatever you want.

A couple of things to be aware of:

- The default elapsed time display gives only minutes and seconds; if the time ever gets up to 60 minutes, it'll “wrap around”, so to speak, and start over again at 0. The same problem can occur if you set the time to over an hour (3600 seconds) with the Set button. However, it's easy to avoid this problem by changing the display to show hours and even days; see TimeFormat, under Customizing, below.
- Don's Up/Down Timer tries to display the correct elapsed time even if it misses one or more seconds; that can happen if the computer or browser is busy and doesn't let it run frequently enough. For example, if the timer doesn't run for 5 seconds, then when it does run, it'll add 5 seconds to or subtract 5 seconds from the displayed time. However, seeing the number skipping seconds is kind of distracting, so it can be turned off; see CanSkip below.

Customizing

To change anything, you'll need to edit one of the .html files or create your own. A .html file is really an ordinary text file, and it can be edited with any text editor, e.g., Windows Notepad or OS X TextEdit. It can also be edited with a word processor like MS Word, but it's better not to because the word processor might add formatting information—perhaps invisible to you!—that can screw things up. While the *program* is in JavaScript, this file is mostly HTML (HyperText Markup Language).

Options

Options are set after the comment "Set the Up/Down Timer options" (surprise, surprise). Any or all can be omitted and the timer will behave reasonably. Currently, the options are:

- **TimeFormat** allows you to format the countdown/count-up display to your liking. For example, instead of the default English, you can use terms from another language; make adjustments like omitting the seconds segment or just using colons between time units; and show any combination of seconds, minutes, hours, and days. "%%D" = days; "%%H" = hours; "%%M" = minutes; "%%S" = seconds. (Default: "%%M min. %%S sec.")

- **FinishMessage**: message to display (or "" for no message) when the countdown reaches zero. (Default: "Time is up!")

- **TimeColor** and **FinishColor** set the color of characters for the time displayed and FinishMessage, respectively. You can use any color name that's legal in JavaScript (I believe the names are exactly the same as in HTML). Lists of names are easy to find on the Web, but "Color Name & Hue" is a very nice interactive tool for choosing colors; it's at

<http://www.colblindor.com/color-name-hue/>

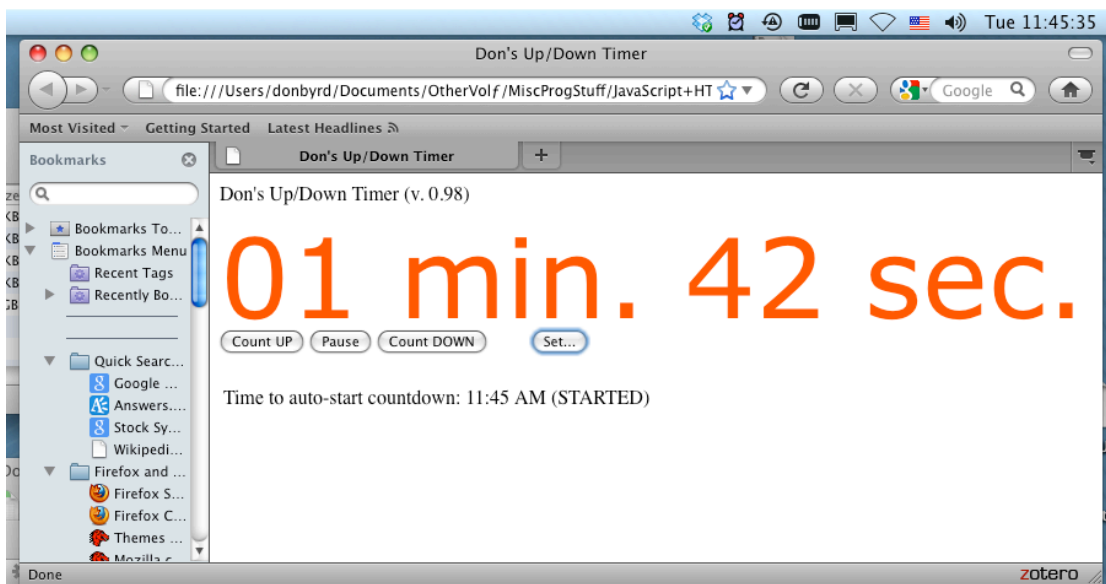
(Defaults: "black", "darkred")

- **FinishSound**: true = play the audio file FinishSound.wav when the FinishMessage is displayed. Caveat: As of this writing, it appears that not all browsers can play .wav audio files, but some that can't can play other formats (e.g., Ogg Vorbis). In general, Firefox on Windows and Mac OS X and Safari on OS X are okay with the .wav; Internet Explorer on Windows isn't. See comments on this in the DonsUpDownTimer.js file. (Default: false)

- **LeadingZero**: true = display leading zeros; false = don't display them. (Default: true)

- **BoldDigits**: true = display digits in bold (other characters displayed aren't affected). (Default: true)

- **CanSkip**: true = time displayed can skip seconds to maintain accuracy. Skipping seconds is useful because it's possible the timer won't be able to update its display for several seconds at a time. That rarely happens in most environments, but, for example, I've seen Firefox 3.6.x on one OS X computer have to skip repeatedly, sometimes by 10 seconds or more. (Default: true)



- **AutoCountdownTime** is a string giving a time of day in 24-hour format at which counting down should begin automatically: for example, "11:45" for 11:45 AM, or "14:30" for 2:30 PM. It's always interpreted as a time in the current day. Therefore, if you launch the program with AutoCountdownTime = "11:45" at 11:53 AM, it will think the time to start auto-countdown has already passed and will start counting down immediately: probably not what you want! (Default: no auto countdown)
- **AutoCountdownColor** sets the color of characters in the time display once auto countdown begins. (Default: "blue")

Other Customizing

Besides the official "options", you can control lots of other things—for example fonts and font sizes—by making obvious changes, and more with a little knowledge of HTML. Look just below the option-setting code in the sample .html files.

Future Improvements

Improvements I have in mind for when I have time include:

- If the count falls behind, instead of either skipping seconds to catch up or letting it stay behind, update the count very quickly until it's up-to-date (which, if counting down, might mean stopping at zero and showing the Finish message).
- When time is up, optionally make the message flash.
- Check and warn user if the time value is too large for the display format.
- Allow entering an AutoCountdownTime directly in the timer window.
- Add a time scale for countdown so, e.g., it can get down to zero in half or 2/3 the time shown.

Suggestions, anyone?