Opening and Closing Windows using JavaScript

Using multiple events and window names to close windows opened by another document

By Jeanine Meyer  jeanine.meyer@purchase.edu  http://faculty.purchase.edu/jeanine.meyer

JavaScript provides facilities to create new windows, with specified content and features. This article describes a way to close the current window along with other windows opened by JavaScript. In my example, I show how to close the windows as a result of two distinct events: clicking on a button or when a video has ended.

It is possible in JavaScript to open a new window using the open method. This method includes parameters for the address (URL), a name, and features, such as location (top and left), dimensions (width and height) and attributes such as having scroll bars. You can search online and easily find out the details. In these notes, I focus on closing windows opened by JavaScript.

Warning: A window that is opened by the viewer (client) and not by JavaScript cannot be closed by JavaScript.

The challenge I was given was an application involving a sequence of windows being opened, say first opens second opens third. The problem was to write JavaScript in the last script that closed the last window and all the intermediate windows. That is, JavaScript code in third would close itself and also second. The intermediate windows in my student's application played video and audio clips. Closing the intermediate windows was a challenge because the last script does not have access to any variables set in the first script and subsequent scripts.

Consider the following statement:

```javascript
windowAhandler = window.open("somescript.html","A");
```

JavaScript attempts to open a new window with the name "A", loading in the html file "somescript.html". If there is NOT already a window with that name, then a new window is opened, with default features. The variable `windowAhandler` can be used later to close the window and, indeed, do other things such as change window location. The variable `windowAhandler` needs to be accessible, which is not the case for code in "somescript.html".

However, let's continue. If there already is a window open with the name "A", then the variable `windowAhandler` still is set, but no new window is open. This means that the code

```javascript
windowAhandler = window.open("","A");
```

can be used AFTER a window is opened in a script other than the one in which the window was opened originally. For my purposes, my code invokes the open method strictly to get at the handler. This is the solution to the problem!

My test example consists of 3 scripts: first.html, second.html and third.html. The example is tested by loading first.html and pushing the buttons in the sequence of scripts. Note: there is nothing in this technique requiring any HTML5 features, but I use that DOCTYPE because I did use button elements and out of general habit. Notice also that I put in an alert statement just to indicate what was happening. Figure 1 shows the windows stacked up on the screen and the alert message.

Clicking on OK does indeed close the second and third window.

The listings of the 3 scripts are shown in Listing 1.
window.open("second.html","b", "left=20,top=50,width=200,height=200");
</script>
</body>
</html>

This is the first window.
<button onClick="opensecond();">Open second </button>
</body>
</html>

second.html
<!DOCTYPE html>
<html>
<head>
<title>Second window</title>
</head>
<script>
function openthird() {
window.open("third.html","c","left=20,top=200,width=300,height=300");
}
</script>
</body>
</html>

This is the second window.
<button onClick="openthird();">Open third </button>
</body>
</html>

third.html. The alert statement can be removed.
<!DOCTYPE html>
<html>
<head>
<title>Third window</title>
</head>
<script>
function closethisone() {
  //the next line gets a handle on any window named "b"
  bwin = window.open("","b");
  //it uses the handle, bwin, to close the window
  //alert to indicate that the b window is present until
  //the bwin.close() statement
  alert("about to close second window named b");
  bwin.close();
  //close current window, the third and named c window
  window.close();
}
</script>
</body>
</html>

This is the third window.
<button onClick="closethisone();">Close up </button>
</body>
</html>

Listing 1: Code for the 3 scripts

This is a toy example and your situation may require a more complex solution. Keep in mind that you can set up events for setting the focus on a window or responding to the event of a window obtaining focus. The critical factor demonstrated here is that the names given to windows in the original invocations of the open method may be used to close or otherwise manipulate windows opened in any script.

Example with video

My second example, besides featuring media of my grand daughter, demonstrates more complex event handling. Specifically, the closing of windows 2 and 3 can be triggered either by the user clicking a button or a video ending. Figure 2 shows the opening screen.

Clicking on the Open second window button brings out the second window for this application. You can move this and the next window around to assure yourself that all windows are open.

Clicking on the Open third window button invokes the html script shown in Figure 4.

Example with video

My second example, besides featuring media of my grand daughter, demonstrates more complex event handling. Specifically, the closing of windows 2 and 3 can be triggered either by the user clicking a button or a video ending. Figure 2 shows the opening screen.

Clicking on the Open second window button brings out the second window for this application. You can move this and the next window around to assure yourself that all windows are open.

Clicking on the Open third window button invokes the html script shown in Figure 4.
Besides demonstrating that the HTML documents for these windows can contain a variety of HTML content, the significance of this example is the way the second and the third window can be closed. I basically used the same techniques as in the simple example, including the definition of the `closethisone` function. What was new in this example was that I set up `closethisone` as the event handler for the video ended event. This is done in an `init` function. Listing 2 shows the html document for the third script. Notice the `init` function and how it is invoked through action of the `onload` attribute in the `<body>` tag.

```html
<!DOCTYPE html>
<html>
<head>
<title>Third window</title>
</head>
<script>
function init() {
    v = document.getElementById("drumming");
    v.addEventListener("ended", closethisone, false);
}
function closethisone() {
    //the next line gets a handle on any window named "b"
    bwin = window.open(" ", "b");
    //use the handle, bwin, to close the b window
    bwin.close();
    //close current window, the third and named c window
    window.close();
}
</script>
</head>
<body onload="init();">
This is the third window. When the video finishes, this window and the second window will be closed OR you can click on the button. <br/>
<video id="drumming" preload="auto" controls="controls" autoplay>
<source src="drumming.webm" type='video/webm; codec="vp8, vorbis">'
<source src="drumming.mp4">
<source src="drumming.ogg" type='video/ogg; codecs="theora, vorbis">'
Your browser does not accept the video tag.
</video>
<button onClick="closethisone();">Close up second and third windows. </button>
</body>
</html>
```

Listing 2: HTML document for the third window

The fact that `closethisone` does double duty, serving as the event handling for the button `onClick` and the video ended, is the technique that makes this application work and is a useful technique for you to understand. By the way, it appears that using the `addEventListener` method is preferable to using something like `v.onended = closethisone;`

The latter approach works for some browsers, but not others.

Learn more

There are many sources, online and in-print and some sort of e-books, for learning standard HTML, and HTML5 and JavaScript techniques. Here are links to my recent books and the website for the applications described in this article.


- HTML5 and JavaScript Projects, [http://www.apress.com/9781430240327](http://www.apress.com/9781430240327). This book is more advanced than the first one. Chapters describe ways to use video and canvas and ways to combine video and canvas, as well as localStorage, use of Google Maps, geolocation, and php and MySQL. Several of the chapters describe ways to build sets of applications.

- To see the basic application in action and to view the source code, go to [http://faculty.purchase.edu/jeanine.meyer/html5/first.html](http://faculty.purchase.edu/jeanine.meyer/html5/first.html) and click on the buttons. For the example which involves the video-ended event (featuring Annika in an image and two video clips), check out [http://faculty.purchase.edu/jeanine.meyer/html5/openclose.html](http://faculty.purchase.edu/jeanine.meyer/html5/openclose.html).

Jeanine Meyer lives just north of New York City and currently teaches at Purchase College/SUNY after many years at IBM, doing research on robotics and manufacturing and consulting on educational grants. She likes providing programming examples for her Mathematics/Computer Science and New Media students and really, really likes working with images and video clips of her granddaughter and other family members.