

Generating HTML

To see how you can generate HTML in JavaScript and add it to a web page, look at the example [genSample.html](#). There is a link at the bottom of the page that will open the JavaScript file, `genSample.js`, in a separate window.

In the HTML file, `genSample.html`, notice the **onload** attribute in the body tag. The value of the onload attribute tells the browser what JavaScript code to execute when it has finished loading the HTML file. In this case, it executes the function call **showCountdown()**. You can see the definition of `showCountdown` in the `genSample.js` file.

The function `showCountdown` begins by calling **document.getElementById**. We will work a lot more with `getElementById` later in the term. The main thing you need to know about it now is that it returns a pointer to an element in the HTML of the document with which the JavaScript code is associated. The parameter to `getElementById` is **countdownDiv**. If you look back at `genSample.html`, you'll see a div element with id of `countdownDiv`. By setting the id attribute of a div, we can make it easily accessible from JavaScript.

In `genSample.html`, the content of the `countdownDiv` says:

You won't see this text in the browser window because it will be overwritten when the countdown HTML is put into this div.

Using `getElementById` to get a pointer to the `countdownDiv` is the first step to modifying the contents of the div and replacing the text shown in the HTML file. The second step is to use the `innerHTML` property to change the contents of the div, which is what happens in the second line of `showCountdown`.

The `innerHTML` property makes it possible to put a string of HTML in as the new contents of an HTML element such as a div or a paragraph. It will be just as if the string was typed in the source file. In the case of the `showCountdown` function, the string that is put in as the new contents of the `countdownDiv` is the return value of `genCountdown`.

The purpose of the **genCountdown** function is to generate the HTML that will be put in as the new content of the `countdownDiv`. The HTML that it generates will be a string that has ten h2 header elements in it. Each h2 element will have an opening h2 tag, including a **class** attribute and quoted value, contents, and a closing h2 tag.

At the beginning of the function is the declaration for a variable named **html**. That variable is used to hold a string that the function builds to be the new contents of the countdown div. The variable is initialized with an empty string as its value. Later in the function, other strings will be appended to the string in the `html` variable to make the countdown HTML that will be put into `countdownDiv`.

There are two other variables in `genCountdown`. One is **i**, which is used as a loop counter and also as the text contents of each h2 element. The other is **hClass**, which is used to hold the value of the class attribute for each header. The initial value for `hClass` is "hot", but we'll see in the loop that that value is overwritten before it is used.

Next in `genCountdown` is a for loop. The loop counter, `i`, is initialized to 10, and decremented each time through the loop. In the body of the for loop is an if-else statement that assigns a value to the `hClass` variable based on the value of the loop counter. When the loop counter is greater than 8, the value of

hClass will be "cold". If you look at the beginning of genSample.html you will see a style sheet with several classes, including .cold, with the text color set to blue. So, the if-else in the for loop will set the value of hClass to "cold" for 10 and 9, "cool" (green) for 8 and 7, "warm" (orange) for 6, 5, and 4, and "hot" (red) for 3, 2, and 1.

After the if-else statement is an assignment statement that appends a string to the html variable. The string that it appends is the HTML for an h2 header, with the class set to the contents of the hClass variable, and the contents (between the opening and closing h2 tags) set to the value of the loop counter, i. Notice the use of back slashes to put quotes into the generated HTML. An alternative approach would be to use single quotes on the value of the class attribute.

After the for loop, an h1 header with the contents "Blast Off!" is appended to the HTML string. Then the html variable is returned as the return value of the genCountdown function.