An Animated Slideshow

What this chapter covers:

- The basics of animation
- Web page enhancement with animation

In this chapter, you’ll see one of the most dynamic applications of CSS-DOM: the ability to animate elements.

Animation basics

The previous chapter introduced the DOM’s ability to update the styles attached to a document. Using JavaScript to add presentational information can save you time and effort, but, for the most part, CSS remains the best tool for the job.

There’s one area, however, where CSS can’t really help much. If you want to change an element’s style information over time, you’ll probably need to use JavaScript. CSS3 introduces a number of different transition methods that allow you to animate elements. However, currently, browser support without vendor-specific prefixes is minimal. JavaScript, on the other hand, allows you to execute functions at set intervals. This means that you can alter an element’s style with the passage of time.

Animation is the perfect example of this kind of change. In a nutshell, animation involves changing an element’s position over time.

Position

An element’s position in the browser window is presentational information. As such, it is usually added using CSS. Here’s an example that sets an element’s position on the page:

```javascript
const element = {  
  position: absolute;  
  top: 50px;  
  left: 100px; 
};
```

That will position the element 100 pixels from the left of the browser window and 50 pixels from the top. Here’s the DOM equivalent of the same information:

```javascript
element.style.position = "absolute";
element.style.left = "100px";
element.style.top = "50px";
```
Valid values for the position property are "static", "fixed", "relative", and "absolute". Elements have a position value of "static" by default, which simply means that they appear one after the other in the same sequence as they occur in the markup. The "relative" value is similar. The difference is that relatively positioned elements can be taken out of the regular flow of the document by applying the float property.

By applying a value of "absolute" to an element's position, you can place the element wherever you want in relation to its container. The container is either the document itself or a parent element with a position of "fixed" or "absolute". It doesn't matter where the element appears in the original markup, because its position will be determined by properties like top, left, right, and bottom. You can set any of those properties using pixels or percentages.

Setting an element's top property will place the element a specified distance from the top of the document. An element's bottom property will place it a specified distance from the bottom of the document. Similarly, left and right can be used to place the element a specified distance from the left and right edges of the document, respectively. It's a good idea to use either top or bottom, but not both. Likewise with left and right.

Positioning an element in the document is relatively straightforward. Say you had an element like this:

```html
<p id="message">Whee!</p>
```

You could set the message element's position in JavaScript with a function like this:

```javascript
function positionMessage() {
    if (!document.getElementById) return false;
    if (!document.getElementById("message")) return false;
    var elem = document.getElementById("message");
    elem.style.position = "absolute";
    elem.style.left = "50px";
    elem.style.top = "100px";
}
```

Calling the positionMessage function when the page loads will position the paragraph 50 pixels from the left and 100 pixels from the top of the browser window:

```javascript
window.onload = positionMessage;
```

Better yet, use the addLoadEvent function:

```javascript
function addLoadEvent(func) {
    var oldonload = window.onload;
    if (typeof window.onload != 'function') {
        window.onload = func;
    } else {
        window.onload = function() {
            oldonload();
            func();
        }
    }
}
addLoadEvent(positionMessage);
```

Here, the element has been positioned absolutely.