Summary

In this lesson, we will summarize what we learned in this chapter.

The HTML markup provides a hierarchy of elements, following the nesting of HTML elements and attributes in the page's markup. The web page is called a document in the HTML standard, and its structure is represented by an abstract hierarchy, called **Document Object Model**, often referred to as *DOM*.

The DOM is a very important representation of the page. It is used by the browser to embody the page in its memory, and also by browser plug-ins to allow queries and changes within this model. From the page creator's point of view, DOM can be used in scripts programmatically to access and alter document elements, and so it is the base of creating interactive web pages.

The DOM is represented as a document tree where HTML elements, texts, and attributes are the nodes of the tree. Each node may have zero, one, or more child nodes; and each child node has exactly one parent node. DOM has a JavaScript API that supports navigation and changes through the document object.

With a few methods like `getElementById()`, `getElementsByTagName()`, `getElementsByName()`, and a number of useful properties, including anchors, forms, images, and links, you can access single nodes representing HTML elements, or a collection of HTML elements. When you grab an element, you

can navigate to its children, siblings, or attributes.

The DOM API provides simple methods to insert, replace, or remove elements in the document tree. You can add event handler methods to the page, and in these methods, you can change the tree, providing interactive user experience without round-tripping to the webserver.