

The DOM Application Programming Interface

In this lesson, we will deal with the DOM application programming interface.
Let's begin!

WE'LL COVER THE FOLLOWING



- The DOM and the application programming interface (API)
- Abstract interfaces

The DOM and the application programming interface (API)

Besides describing a standard object model for HTML, the DOM standard defines an **Application Programming Interface (API)** to manage this model.

The available functionality of this API is defined as:

*a set of abstract interfaces, such as **HTMLDocument** (describes the behavior of an HTML document behind a web page), **HTMLElement** (a single instance of a markup element within a document), **HTMLCollection** (a collection of **HTMLElement** instances), and others.*

These abstract interfaces are implemented in multiple places to support all DOM related operations.

Abstract interfaces

First, the browser implements them so that it can display the web page based on the HTML markup requested from a server.

Often the browser prefers this interface to plug ins so that they can leverage

Often the browser prefers this interface to plug-ins so that they can leverage the DOM to analyze and transform a loaded page.

A good example of a plug-in analyzing the page is the HTML5 Outliner tool you met in Chapter 4.

Skype and other VoIP apps often install browser plug-ins that recognize phone numbers in the web page and surround them with a context-specific markup that allows starting calls. These use the DOM to *transform* the loaded web page.

Second, these abstract interfaces are implemented in the JavaScript engines of browsers so that page developers will be able to write scripts interacting with this model.

In this course, I will cover only the JavaScript implementation of DOM interfaces (partially), and will not treat plug-in development, which is very *browser-specific*.

The HTML DOM API standardizes the properties and operations that can be used on specific objects of the model.

Nonetheless, many browser vendors extend this API with **extra operations**. These may provide a simpler way to carry out certain tasks, but you may create a page that does not work on every browser.

Now that we've got this covered, in the *next lesson*, we'll see what the DOM exactly covers.

See you there! :)