Dylan Warner

6/19/2021

WDD 330

W09 Readings

**Chapter 9: The Window Object**

*The Browser Object Model -*

* The Browser Object Model (or BOM for short) is a collection of properties and methods that contain information about the browser and computer screen.
* Every browser window, tab, popup, frame, and iframe has a window object.
* Dialogs - alert(), confirm(), prompt() are all methods of the window object.
* Browser Information:
* Window.location- an object that contains information about the URL of the current page.
* Window.location.href returns the full URL as a string.
* Window.location.href can also be written to which will have the page navigate to the assigned URL.
* The *protocol* property returns a string describing the protocol used.
* The *host* property returns a string describing the domain of the current URL and the port number (this is often omitted if the default port 80 is used).
* The *hostname* property returns a string describing the domain of the current URL.
* The *port* property returns a string describing the port number, although it will return an empty string if the port is not explicitly stated in the URL.
* The *pathname* property returns a string of the path that follows the domain.
* The *search* property returns a string that starts with a '?' followed by the query string parameters. It returns an empty string if there are no query string parameters.
* The *hash* property returns a string that starts with a "#" followed by the fragment identifier. It returns an empty string if there is no fragment identifier.
* The *origin* property returns a string that shows the protocol and domain where the current page originated from. This property is read-only.
* The *reload*() method can be used to force a reload of the current page. If it’s given a parameter of true, it will force the browser to reload the page from the server, instead of using a cached page.
* The replace() method is almost the same as the assign() method, except the current page will not be stored in the session history, so the user will be unable to navigate back to it using the back button.
* The *window.history* property can be used to access information about any previously visited pages in the current browser session.
* The *window.history.go()* method can be used to go to a specific page, where 0 is the current page.
* A new window can be opened using the *window.open*() method. This takes the URL of the page to be opened as its first parameter, the window title as its second parameter, and a list of attributes as the third parameter.
* The *window.screen* object contains information about the screen the browser is displayed on.
* The *availHeight* and *availWidth* can be used to find the height and width of the screen, excluding any operating system menus.

*The Document Object -*

* Each window object contains a document object. This object has properties and methods that deal with the page that has been loaded into the window.
* The *write*() method simply writes a string of text to the page. If a page has already loaded, it will completely replace the current document.

*Timing Functions -*

* The *window.setTimeout*() method accepts a callback to a function as its first parameter and a number of milliseconds as its second parameter.
* The *window.setInterval*() method works in a similar way to *window.setTimeout*(), except that it will repeatedly invoke the callback function after every given number of milliseconds.

**Chapter 14: HTML5 APIs**

*HTML5 APIs -*

* The Web Storage API provides a key-value store on the client’s computer that is similar to using cookies but has fewer restrictions, more storage capacity, and is generally easier to use.
* If a browser supports the Web Storage API, the window object will have a property called localStorage, which is a native object with a number of properties and methods used to store data.

EX.

*localStorage.setItem('name', 'Walter White');*

*localStorage.getItem('name');*

*<< "Walter White"*

*localStorage.name = 'Heisenberg'; console.log(localStorage.name);*

*<< "Heisenberg";*

To remove an entry from local storage, use the removeItem method:

*localStorage.removeItem('name');*

Alternatively, this can be done using the delete operator:

*delete localStorage.name;*

To completely remove everything stored in local storage, use the clear() method:

*localStorage.clear();*

* Geolocation API - The Geolocation API is used to obtain the geographical position of the device. This means it can be used to find the user’s exact location, then link to nearby places or measure the speed at which the user is moving.
* Web Workers API - We saw in earlier chapters that JavaScript is a single-threaded language, meaning that only one process can run at one time. Web workers allow processes to be run in the background, adding support for concurrency in JavaScript.
* Websockets - Websocket is a new protocol that allows two-way communication with a server – also known as push messaging. This means that a connection is kept open and responses are ‘pushed’ to the client as soon as they are received.
* Notifications - The Notification API allows you to show messages using the system's notifications. This is usually a popup in the corner of the screen, but it changes depending on the operating system.
* Multimedia - HTML5 introduced the <audio> and <video> tags used to insert audio and video clips into a web page. It also introduced a Media API for controlling the playback of the clips using JavaScript.