**Document Object Model**

**Objects**

Window

Methods:

Window.alert(‘Hello’) – Displays an alert message at the top of the tab saying “Hello”.

Window.open(‘https://www.google.com’); - Opens a new tab and loads “google.com”.

Window.scrollBy({  
 top: window.innerHeight, // The length of the visible page

Left: 0, // Do not scroll to the left

Behavior: ‘smooth’ // Animate the scrolling smoothly as opposed to ‘instant’ or ‘auto’

}); - Scrolls down the webpage.

Window.location.reload() – Reloads the webpage.

Window.localStorage.setItem(‘test’, 20) – Creates a “Key-Value” pair of “test-20” and stores the same in the browser. (Under “Application” as opposed to “console”)

Window.localStorage.getItem(‘test’) – Returns 20.

Properties

innerHeight – Height in pixels of the visible web page.

outerHeight – Height in pixels of the entire browser.

scrollX – Width of the current position of the scrollbar (at the left of the bar).

scrollY – Height of the current position of the scrollbar (at the top of the bar).

innerWidth – Width of the visible web page.

Document.body.scrollHeight – Height of the entire webpage in pixels.

Event Listeners

Window.onscroll = function () {

Console.log(“Hi”);

} – Prints “Hi” to the console every time the scroll bar is moved.

Given a variable containing a specific element (assigned via the querySelector method), there are three ways to assigned an event listener, say the “onclick” event listener, to this variable via javascript:

1. btn.addEventListener(‘click’, addParagraph); – ‘click’ event invokes the “addParagraph” function that is user-defined.
2. btn.onclick = addParagraph();
3. On the HTML element itself.

Document

To Create a new element using DOM

1. Create variables to store the various nodes in the element. Nodes refers to the individual pieces of the element declaration. For example, a paragraph that says “something”, is declared in HTML as: <p class=”some-class”>something</p>. Here the nodes include: the “p”- the element node, “class” – the attribute node, and “something” – the text node.
2. Append the children elements to their parents and grandparents.

For example:

const elementNode = document.createElement(‘p’);

const attributeNode = document.createElement(‘class’);

const textNode = document.createElement(‘something’);

attributeNode.value = ‘some-class’;

elementNode.setAttibuteNode(attributeNode);

elementNode.appendChild(textNode);

document.body.appendChild(elementNode);

The above programmatically creates a new HTML paragraph tag, and appends it to a webpage.

Other Methods

document.querySelector(‘button’) – Selects the first button within a webpage.

document.querySelectorAll(‘button’) – Selects all buttons within a webpage.

#... – Refers to the id of an element.

.\_\_ - Refers to the class of an element

document.getElementsById(‘someElement’) - Select the element specified by the Id. Equivalent to “document.querySelector(#someElement)”

someElement.children.item(index) – Returns the child node of the element “someElement” specified at index “index” specified.

Node Elements

someNode.textContent = ‘some string’; – Appends some text to some created element. It is a shorter process than the above under “To Create a new element using DOM”. An equivalent of this would be: *someNode.innerHTML = ‘some string’;*

someNode.nodeType – Returns the type of the node in the form of an index. E.g. 1 represents “elements”, 2-“attributes” etc.

someParentNode.insertBefore(newElement, elementBeingPushedDown) – Inserts a new element before another which already exists.

N/B: Both Window and Document are global objects and so their methods and properties can be invoked without the object name in “objectName.method()”