

CSE 154: Web Programming

Lecture 8 JS “Cheat Sheet”

This reference summarizes the most useful methods/properties used so far CSE 154. It is not an exhaustive reference for everything in JavaScript (for example, there exist many more window methods/properties than are shown below), but provide most functions/properties we have seen so far.

The Module Pattern

Whenever writing JavaScript, you should use the module pattern, wrapping the content of the code (window load event handler and other functions) in an anonymous function. Below is a template for reference:

```
“use strict”;
(function() {
  // any module-globals (limit the use of these when possible)
  window.addEventListener("load", init);

  function init() {
    ...
  }

  // other functions
})();
```

Handy Alias Functions

The following four shorthand functions will be used frequently in the class (to use, make sure they are defined in the module pattern of your JS program).

```
function id(idName) {
  return document.getElementById(idName);
}

function qs(selector) {
  return document.querySelector(selector);
}

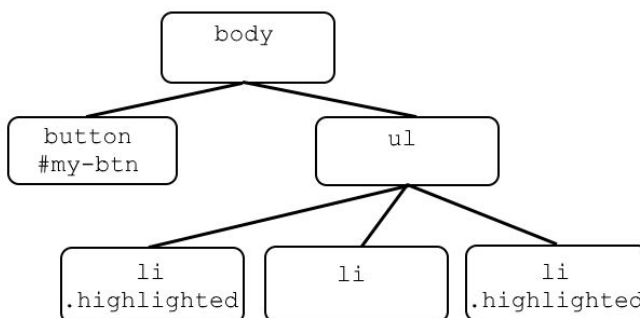
function qsa(selector) {
  return document.querySelectorAll(selector);
}

function gen(elType) {
  return document.createElement(elType);
}
```

Accessing DOM elements from the document

These are methods/properties than can be accessed from the global document object, for example:

```
document.getElementById("my-btn");
```



Method/Property and Example	Description
getElementById(idName) getElementById("my-btn");	Returns a DOM object whose id property matches the specified string. If no matches are found, null is returned.
querySelectorAll(selector) querySelectorAll("li.highlighted");	Returns a list of the document's elements that match the specified group of selectors. If no matches are found, null is returned.
querySelector(selector) querySelector("li.highlighted");	Returns the first DOM element that matches the specified selector, or group of selectors. If no matches are found, null is returned.

DOM Element .classList Methods

Method	Description
el.classList.add(class) div.classList.add("skittle") div.classList.add("skittle", "green");	Adds specified class values. These values are ignored if they already exist in the list.
el.classList.remove(class) div.classList.remove("green");	Removes the specified class value
el.classList.toggle(class) div.classList.toggle("hidden");	Toggles the listed class value. If the class exists, then removes it and returns false, if it did not exist in the list add it and return true
el.classList.contains(class) div.classList.contains("highlighted");	Returns true if the specified class value is exists in the classList for this element

DOM Element Methods and Properties

Recall that if you have an HTML element on your page that has attributes, you can set those properties through JavaScript as well. For instance if your

```

```

You could do the following in your JavaScript code (using the `id` alias for `document.getElementById`):

```
id("dog-tag").alt = "My really cute dog";
```

Example DOM Element attributes include (other than `src`, and `alt` above) are:

Property	Description
<code>el.disabled</code>	Whether or not the DOM element (e.g. a button or input) is disabled
<code>el.value</code>	The value attribute of form elements (input, textarea, checkbox radio, select, etc.)
<code>el.id</code>	The id attribute of an element
<code>el.textContent</code>	Sets or returns the text content of the specified node
<code>el.innerHTML</code>	Sets or returns the HTML content of an element
<code>el.getAttribute(attr)</code>	Returns the specified attribute value <code>attr</code> of <code>e1</code>
<code>el.children</code>	Returns a collection of the child elements of <code>e1</code>
<code>el.parentNode</code>	Returns the parent node of <code>e1</code>
<code>el.classList</code>	Returns the class name(s) of <code>e1</code>
<code>el.className</code>	Sets or returns the value of the class attribute of <code>e1</code>

DOM Manipulation Methods

Method/Property and Example	Description
<code>document.createElement(tagname)</code> <code>let li = document.createElement("li");</code>	Creates and returns an Element node. Note that this method is used on document not a DOM node.
<code>e1.appendChild(child)</code> <code>ol.appendChild(li);</code>	Adds a new child node to e1 as the last child node
<code>e1.removeChild(child)</code> <code>ol.removeChild(li);</code>	Removes a child node from an element
<code>e1.insertBefore(newNode, refNode);</code> <code>ol.insertBefore(newLi, existingLi);</code>	Adds newNode to parent e1 before e1's child refNode position

DOM and Events

DOM Method/Property	Description
<code>e1.addEventListener(event, fn)</code>	Attaches an event handler function fn to the specified element e1 to listen to event
<code>e1.removeEventListener(event, fn)</code>	Removes the event handler fn to the specified e1 listening to event

Event Types

click	mousemove	keydown	change
dblclick	mouseout	error	focus
mouseenter	mouseover	success	submit
mouseleave	mouseup	load	select
mousedown	keyup	unload	resize

Useful Event Object Methods and Properties

Any function assigned in an `addEventListener` can accept an optional argument, which will be the Event object. These are some things you can do with that object.

```
function init() {
  id("my-btn").addEventListener("click", handleClick);
}

function handleClick(evt) {
  console.log(evt.target); // #my-btn
}
```

Method	Description
<code>evt.target</code>	Returns the element that triggered the event
<code>evt.type</code>	Returns the name of the event
<code>offsetX</code>	Returns the horizontal coordinate of the mouse pointer, relative to the DOM element clicked
<code>offsetY</code>	Returns the vertical coordinate of the mouse pointer, relative to the DOM element clicked
<code>timestamp</code>	Timestamp (in ms) the event object was created.

JavaScript string Methods and Properties

Method	Description
<code>length</code>	Returns the length of a string
<code>charAt(index)</code>	Returns the character at the specified index
<code>indexOf(string)</code>	Returns the position of the first found occurrence of a specified value in a string
<code>split(delimiter)</code>	Splits a string at instances of the delimiter into an array of substrings
<code>substring(start, end)</code>	Extracts the characters from a string between two specified indices
<code>trim()</code>	Removes whitespace from both ends of a string
<code>toLowerCase()</code>	Returns a lowercase version of a string
<code>toUpperCase()</code>	Returns an uppercase version of a string

JavaScript Array Methods and Properties

Method	Description
length	Sets or returns the number of elements in an array
push(e1)	Adds new elements to the end of an array and returns the new length
pop()	Removes and returns the last element of an array
unshift(e1)	Adds new elements to the beginning of an array and returns the new length
shift()	Removes and returns the first element in an array
sort()	Sorts the elements of an array
join()	Returns a string concatenating all elements of an array (maintaining order)
indexOf(e1)	Returns the index of the element in the array, or -1 if not found

JavaScript Math Functions

Method	Description
Math.random()	Returns a double between 0 (inclusive) and 1 (exclusive)
Math.abs(n)	Returns the absolute value of n
Math.min(a, b, ...)	Returns the smallest of 0 or more numbers
Math.max(a, b, ...)	Returns the largest of 0 or more numbers
Math.round(n)	Returns the value of n rounded to the nearest integer
Math.ceil(n)	Returns the smallest integer greater than or equal to n
Math.floor(n)	Returns the largest integer less than or equal to n
Math.pow(n, e)	Returns the base n to the exponent e power, that is, n^e