

DOM Selectors

<code>\$('#id')</code>	<code>document.getElementById('id')</code>
<code>\$('tagName')</code>	<code>document.getElementsByTagName('tagName')</code> <code>document.querySelectorAll("tagName") //returns a NodeList</code>
<code>\$('.className')</code>	<code>document.getElementsByClassName('className')</code>
<code>\$('tagName.className')</code>	<code>document.querySelector('tagName.className');</code>
<code>\$('#id').text('text')</code>	<code>document.getElementById(id).innerHTML = "text"</code>
<code>\$('#id').attr('attr', 'new value')</code>	<code>document.getElementById(id).attribute = new value</code>
<code>.attr ^ or removeAttr()</code>	<code>document.getElementById("id").setAttribute("attr", "val"); // or getAttribute</code>
<code>\$('#id').css({color:red})</code>	<code>Object.assign(document.getElementById('id').style,{fontSize:"20px",color: red});</code>
<code>\$("[attribute='value']")</code>	<code>document.querySelectorAll("[attribute='value']");</code> IE9+

In POJS A NodeList object is a list (collection) of nodes extracted from a document

Events

<code>\$('#id').click(function(){})</code>	<code>document.getElementById("id").onclick =function(){}</code>
<code>\$('#id').on('event', function(){})</code>	<code>element.addEventListener('click', function() { ... }, false);</code>
<code>\$('#id').on('mouseover click', function () { })</code>	<code>['click', 'mouseover'].map(function(e) { window.addEventListener(e, mouseMoveHandler); });</code>
<code>\$.ready(function () { });</code>	<code>document.addEventListener("DOMContentLoaded", function(event) { })</code> IE9+
<code>\$(window).load(function(){ });</code>	<code>window.onload = doSomething or function(){ };</code>
<code>\$('#image').load(function(){ });</code>	<code>imageObj.addEventListener("load", function() { }, false);</code>

ready and DOMContentLoaded are executed as soon as the HTML has downloaded and the DOM is ready, but it won't wait for other assets.

`$(window).load` and `window.onload` are triggered after all external resources (like images) have finished loading



By **dwapi**
cheatography.com/dwapi/

Not published yet.
Last updated 13th October, 2017.
Page 1 of 3.

Sponsored by **Readable.com**
Measure your website readability!
<https://readable.com>

Enumerables

<code>\$.merge([0, 1], [2, 3])</code>	Joins two or more arrays, and returns a copy of the joined arrays	<code>[0, 1].concat([2, 3])</code>
<code>\$.each([52, 97], function(index, item){})</code>	Calls a function for each array element	<code>array.forEach(function(currentValue, index, arr), function({})</code>
<code>\$.grep([0, 1], function(item, index){})</code>	Returns a new array with matching elements	<code>[1,2].filter(function(currentValue, index, arr){})</code>
<code>\$.map(array, function(item, index) { return doSomething });</code>	Creates a new array with the result of calling a function on each array element	<code>array.map(function(currentValue, index, arr){ return doSomething; })</code>
<code>\$.unique(array)</code>	Remove duplicates from an array	<code>arr.filter(function(item, index, inputArray) { return inputArray.indexOf(item) == index; });</code>
N/A	Reduce the values of an array to a single value (going left-to-right)	<code>[1, 2, 3].reduce(function(total, num){ return total + num; })</code>
N/A	Reduce the values of an array to a single value (going right-to-left)	<code>[1, 2, 3].reduceRight(function(total, num){ return total + num; })</code>
N/A	Removes the first element of an array, and returns that element	<code>[1, 2, 3].shift()</code>
N/A	Adds new elements to the beginning of an array, and returns the new length	<code>[1, 2, 3].unshift(4, 5)</code>
N/A	Add items to the array based on index (rather than shift or push)	<code>[1, 2, 3, 4].splice(2, 0, "a", "b") // [1, 2, "a", "b", 3, 4]</code>



By **dwapi**
cheatography.com/dwapi/

Not published yet.
 Last updated 13th October, 2017.
 Page 2 of 3.

Sponsored by **Readable.com**
 Measure your website readability!
<https://readable.com>

DOM Manipulation

<code>var newElement = \$("<div>", {"class": "something"});</code>	<code>var newElement = document.createElement('div',{ class: 'something'});</code>
<code>\$(newElement).text('text')</code>	<code>var node = document.createTextNode("This is a new paragraph."); newElement.appendChild(node);</code>
<code>\$("#parentNode").append(\$(newElement))</code>	<code>var parentNode = document.getElementById("div1"); element.appendChild(para);</code>
<code>\$("#parentNode").prepend(\$(newElement))</code>	<code>parentNode.insertBefore(newElement, parentNode.firstChild);</code>
<code>\$(newElement).before(\$("#firstChild"));</code>	<code>parentNode.insertBefore(newElement, parentNode.firstChild);</code>
<code>\$("#inner").wrap(\$(outer));</code>	<code>document.getElementById("outer").innerHTML = raw_html or newElement;</code>
<code>\$("#parentNode").remove()</code>	<code>parentNode.removeChild(child);</code>
<code>\$("#oldNode").replaceWith(\$("#newNode"))</code>	<code>parentNode.replaceChild(newNode, oldNode);</code>
<code>\$("#id").clone()</code>	<code>document.getElementById("id").cloneNode(true);</code>
<code>\$("#wrap").is("empty")</code>	<code>document.getElementById("id").childNodes()</code>
<code>\$("#id").addClass('newClass') or removeClass()</code>	<code>document.getElementById("myelement").classList.add("newClass"); or .remove()</code> IE10+
<code>\$("#id").hasClass('myclass')</code>	<code>document.getElementById("myelement").classList.contains("myclass");</code> IE110+
<code>\$("#id").toggleClass('myclass')</code>	<code>document.getElementById("myelement").classList.toggle("myclass");</code> IE110+

toggleClass and toggle will remove a class if present, or add it if not



By **dwapi**
cheatography.com/dwapi/

Not published yet.
Last updated 13th October, 2017.
Page 3 of 3.

Sponsored by **Readable.com**
Measure your website readability!
<https://readable.com>