

## Celestra v1.18.1 FP cheatsheet – <https://github.com/Serrin/Celestra/>

Basic API	DOM	Functional programming
<code>doc = document</code> <code>qsa(&lt;selector&gt;[,context]).each( fn (el,i) { el.arguments;} );</code> <code>qs(&lt;selector&gt;[,context]).argument;</code> <code>domReady(&lt;fn&gt;);</code> <code>inherit(&lt;subclass&gt;,&lt;superclass&gt;);</code> <code>random(&lt;max&gt;); or random(&lt;min&gt;,&lt;max&gt;);</code> <code>getScript(&lt;url&gt;[,success]);</code> <code>getScripts(&lt;scripts&gt;);</code> <code>getStyle(&lt;href&gt;[,success]);</code> <code>getStyles(&lt;styles&gt;);</code> <code>getUrlVar([name]);</code> <code>getUrlVarFromString(&lt;querystr&gt;[,name]);</code> <code>obj2string(&lt;object&gt;);</code> <code>getType(&lt;variable&gt;[,type]);</code> <code>extend([deep,]&lt;target&gt;,&lt;source1&gt;[,srcN]);</code> <code>getFullscreen();</code> <code>setFullscreenOn(&lt;selector&gt; or &lt;element&gt;);</code> <code>setFullscreenOff();</code> <code>getLocation(&lt;success&gt;[,error]);</code> <code>getDoNotTrack();</code> <code>form2array(&lt;form&gt;);</code> <code>form2string(&lt;form&gt;);</code> <code>constant(&lt;value&gt;);</code> <code>identity(&lt;value&gt;);</code> <code>noop();</code> <code>repeat(&lt;iteration&gt;,&lt;callback&gt;);</code>	<code>domCreate(&lt;type&gt;[,properties[,innerHTML]]);</code> <code>domGetCSS(&lt;element&gt;,&lt;property&gt;);</code> <code>domSetCSS(&lt;element&gt;,&lt;property&gt;,&lt;value&gt;);</code> <code>domSetCSS(&lt;element&gt;,&lt;properties&gt;);</code> <code>domFadeIn(&lt;element&gt;[, duration[,display]]);</code> <code>domFadeOut(&lt;element&gt;[,duration]);</code> <code>domFadeToggle(&lt;element&gt;[,duration[,display]]);</code> <code>domShow(&lt;element&gt;[,display]);</code> <code>domHide(&lt;element&gt;);</code> <code>domToggle(&lt;element&gt;[,display]);</code> <code>domOn(&lt;eventTarget&gt;,&lt;eventType&gt;,&lt;callback&gt;);</code> <code>domOff(&lt;eventTarget&gt;,&lt;eventType&gt;,&lt;callback&gt;);</code> <code>domTrigger(&lt;eventTarget&gt;,&lt;eventType&gt;);</code>	<code>toFunction(&lt;function&gt;);</code> <code>bind(&lt;function&gt;,&lt;context&gt;);</code> <code>forEach(&lt;collection&gt;,&lt;callback&gt;);</code> <code>each(&lt;collection&gt;,&lt;callback&gt;);</code> <code>map(&lt;collection&gt;,&lt;callback&gt;);</code> <code>forIn(&lt;object&gt;,&lt;callback&gt;);</code> <code>mapIn(&lt;object&gt;,&lt;callback&gt;);</code> <code>toArray(&lt;object&gt;);</code> <code>toObject(&lt;array&gt;);</code>
<b>AJAX and CORS</b>		
	<code>getJSON(&lt;url&gt;,&lt;success&gt;);</code> <code>getText(&lt;url&gt;,&lt;success&gt;);</code> <code>getAjax(&lt;url&gt;,&lt;format&gt;,&lt;success&gt;[,error][,user&lt;,&amp;password&gt;]);</code> <code>postAjax(&lt;url&gt;,&lt;data&gt;,&lt;format&gt;,&lt;success&gt;[,error][,user&lt;,&amp;password&gt;]);</code> <code>getCors(&lt;url&gt;,&lt;format&gt;,&lt;success&gt;[,error][,user&lt;,&amp;password&gt;]);</code> <code>postCors(&lt;url&gt;,&lt;data&gt;,&lt;format&gt;,&lt;success&gt;[,error][,user&lt;,&amp;password&gt;]);</code>	
<b>Cookie</b>		
	<code>setCookie(&lt;name&gt;,&lt;value&gt;[,hours[,path[,domain[,secure[,HttpOnly]]]]]);</code> <code>getCookie([name]);</code> <code>hasCookie(&lt;name&gt;);</code> <code>removeCookie(&lt;name&gt;[,path[,domain[,secure[,HttpOnly]]]]]);</code>	
<b>Type checking</b>		
<code>isString(&lt;value&gt;;, isChar(&lt;value&gt;;, isNumber(&lt;value&gt;;, isInteger(&lt;value&gt;;, isFloat(&lt;value&gt;;, isBoolean(&lt;value&gt;;,</code> <code>isObject(&lt;value&gt;; isEmptyObject(&lt;value&gt;;, isFunction(&lt;value&gt;;, isArray(&lt;value&gt;;, isEmptyArray(&lt;value&gt;;,</code> <code>isArraylike(&lt;value&gt;;, isNull(&lt;value&gt;;, isUndefined(&lt;value&gt;;, isNullOrUndefined(&lt;value&gt;;, isPrimitive(&lt;value&gt;;,</code> <code>isSymbol(&lt;value&gt;; ES6, isMap(&lt;value&gt;; ES6, isSet(&lt;value&gt;; ES6, isDate(&lt;value&gt;;, isRegExp(&lt;value&gt;;, isElement(&lt;value&gt;;</code>		
<b>Polyfills</b>		
<code>Array.from(), Array.of(), Object.create(), Object.assign(), ChildNode.after(), ChildNode.before(), ChildNode.remove(),</code> <code>ChildNode.replaceWith(), ParentNode.append(), ParentNode.prepend(), Array.prototype.includes(), String.prototype.includes(),</code> <code>NodeList.prototype.forEach(), Number.MIN_SAFE_INTEGER, Number.MAX_SAFE_INTEGER, Number.EPSILON, Number.isNaN(), isNaN(),</code> <code>Number.isInteger(), Number.isFinite(), Number.isSafeInteger()</code>		