

# Celestra v1.17.1 FP cheatsheet – <https://github.com/Serrin/Celestra/>

Basic API	DOM	Type checking
<code>doc = document</code> <code>qsa(&lt;selector&gt;[, context]);</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;);</code> <code>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;, ...sources);</code> <code>getFullscreen();</code> <code>setFullscreenOn(&lt;selector&gt;);</code> <code>setFullscreenOn(&lt;element&gt;);</code> <code>setFullscreenOff();</code> <code>getLocation(&lt;success&gt;[, error]);</code> <code>getDoNotTrack();</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>isString(&lt;value&gt;);</code> <code>isChar(&lt;value&gt;);</code> <code>isNumber(&lt;value&gt;);</code> <code>isInteger(&lt;value&gt;);</code> <code>isFloat(&lt;value&gt;);</code> <code>isBoolean(&lt;value&gt;);</code> <code>isObject(&lt;value&gt;);</code> <code>isEmptyObject(&lt;value&gt;);</code> <code>isFunction(&lt;value&gt;);</code> <code>isArray(&lt;value&gt;);</code> <code>isEmptyArray(&lt;value&gt;);</code> <code>isArraylike(&lt;value&gt;);</code> <code>isNull(&lt;value&gt;);</code> <code>isUndefined(&lt;value&gt;);</code> <code>isNullOrUndefined(&lt;value&gt;);</code> <code>isPrimitive(&lt;value&gt;);</code> <code>isSymbol(&lt;value&gt;); ES6</code> <code>isMap(&lt;value&gt;); ES6</code> <code>isSet(&lt;value&gt;); ES6</code>
	<b>AJAX and CORS</b>	<b>Functional programming</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;, password&gt;]);</code> <code>postAjax (&lt;url&gt;, &lt;data&gt;, &lt;format&gt;, &lt;success&gt; [, error] [, user&lt;, password&gt;]);</code> <code>getCors (&lt;url&gt;, &lt;format&gt;, &lt;success&gt;[, error] [, user&lt;, password&gt;]);</code> <code>postCors (&lt;url&gt;, &lt;data&gt;, &lt;format&gt;, &lt;success&gt; [, error] [, user&lt;, password&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>Polyfills</b>		
<code>Array.from(), Array.of(), Object.create(), Object.assign(), ChildNode.after(), ChildNode.before(), ChildNode.remove(), ChildNode.replaceWith(), ParentNode.append(), ParentNode.prepend(), Array.prototype.includes(), String.prototype.includes(), NodeList.prototype.forEach(), Number.MIN_SAFE_INTEGER, Number.MAX_SAFE_INTEGER, Number.EPSILON, Number.isNaN(), isNaN(), Number.isInteger(), Number.isFinite(), Number.isSafeInteger()</code>		