

## Celestra cheatsheet – v2.0.5 – <https://github.com/Serrin/Celestra/>

The `celestra` and/or the `_` objects contain these functions, except the polyfills. Example: `_.qsa("p");`

Core API	DOM	Functional programming
<code>qsa(&lt;selector&gt;[,context]).each( function (el, i) { el.arguments; } );</code> <code>qs(&lt;selector&gt;[,context]).argument;</code> <code>domReady(&lt;function&gt;);</code> <code>inherit(&lt;subclass&gt;,&lt;superclass&gt;);</code> <code>random([max]);</code> <code>random(&lt;min&gt;,&lt;max&gt;);</code> <code>randomString([length[,specChar]]);</code> <code>b64Encode(&lt;string&gt;);</code> <code>b64Decode(&lt;string&gt;);</code> <code>javaHash(&lt;data&gt;[,hexa]);</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>deepAssign(&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>removeTags(&lt;string&gt;);</code> <code>createFile(&lt;filename&gt;,&lt;content&gt;[,dType]);</code> <code>fromEntries(&lt;entries&gt;);</code> <code>noConflict();</code> <code>celToWindow();</code> <code>version;</code>	<code>domCreate(&lt;type&gt;[,properties[,innerHTML]]);</code> <code>domCreate(&lt;element descriptive object&gt;);</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> <code>hasOwn(&lt;object&gt;,&lt;property&gt;);</code>
AJAX and CORS		
	<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>getJSON(&lt;url&gt;,&lt;success&gt;);</code> <code>getText(&lt;url&gt;,&lt;success&gt;);</code>	
Cookie		
	<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>	
Type checking		
	<code>isString(&lt;value&gt;;, isChar(&lt;value&gt;;, isNumber(&lt;value&gt;;, isInteger(&lt;value&gt;;,</code> <code>isFloat(&lt;value&gt;;, isNumeric(&lt;value&gt;;, isBoolean(&lt;value&gt;;, isObject(&lt;value&gt;;,</code> <code>isEmptyObject(&lt;value&gt;;, isFunction(&lt;value&gt;;, isArray(&lt;value&gt;;,</code> <code>isEmptyArray(&lt;value&gt;;, isArraylike(&lt;value&gt;;, isNull(&lt;value&gt;;,</code> <code>isUndefined(&lt;value&gt;;, isNullOrUndefined(&lt;value&gt;;, isPrimitive(&lt;value&gt;;,</code> <code>isRegexp(&lt;value&gt;;, isDate(&lt;value&gt;;, isElement(&lt;value&gt;;,</code> <b>ES6:</b> <code>isSymbol(&lt;value&gt;;, isMap(&lt;value&gt;;, isSet(&lt;value&gt;;,</code>	

## Polyfills

**Array:** Array.from();, Array.of();, Array.prototype.fill();, Array.prototype.find();, Array.prototype.findIndex();, Array.prototype.includes();, Array.prototype.flat();, Array.prototype.flatMap();

**String:** String.prototype.includes();, String.prototype.trimStart()/trimLeft();, String.prototype.trimEnd()/trimRight();, String.prototype.startsWith();, String.prototype.endsWith();, String.prototype.padStart();, String.prototype.padEnd();, String.prototype.repeat();

**Object:** Object.create();, Object.assign();, Object.entries();, Object.values();, Object.is();

**DOM:** NodeList.prototype.forEach();, ChildNode.after();, ChildNode.before();, ChildNode.remove();, ChildNode.replaceWith();, ParentNode.append();, ParentNode.prepend();

**Number:** Number.MIN\_SAFE\_INTEGER; , Number.MAX\_SAFE\_INTEGER; , Number.EPSILON; , Number.isInteger();, Number.isSafeInteger();, Number.isFinite();, Number.isNaN();, isNaN();