

## Celestra cheatsheet – v2.5.2 – <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]).forEach(&lt;fn&gt;);</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>importScript(&lt;url&gt;[,success]);</code> <code>importScripts(&lt;scripts&gt;);</code> <code>importScripts(&lt;script1&gt;[,scriptN]);</code> <code>importStyle(&lt;href&gt;[,success]);</code> <code>importStyles(&lt;styles&gt;);</code> <code>importStyles(&lt;style1&gt;[,styleN]);</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>merge([deep,]&lt;target&gt;,&lt;source1&gt;[,srcN]);</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>strRemoveTags(&lt;string&gt;);</code> <code>strReverse(&lt;string&gt;);</code> <code>createFile(&lt;filename&gt;,&lt;content&gt;[,dType]);</code> <code>noConflict();</code> <code>version;</code>	<code>domCreate(&lt;type&gt;[,properties[,innerHTML]]);</code> <code>domCreate(&lt;element descriptive object&gt;);</code> <code>domToElement(&lt;htmlString&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>domSiblings(&lt;element&gt;);</code>	<code>toFunction(&lt;function&gt;);</code> <code>bind(&lt;function&gt;,&lt;context&gt;);</code> <code>hasOwn(&lt;object&gt;,&lt;property&gt;);</code> <code>tap(&lt;value&gt;,&lt;callback&gt;);</code> <code>forEach(&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>constant(&lt;value&gt;);</code> <code>identity(&lt;value&gt;);</code> <code>noop();</code> <code>T();</code> <code>F();</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> <b>Ajax shorthands:</b> <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> and <code>hasCookie(&lt;name&gt;);</code> <code>removeCookie(&lt;name&gt;[,path[,domain[,secure[,HttpOnly]]]]];</code>	
Type Checking		
	<b>ES5:</b> <code>isEqual(&lt;value1&gt;,&lt;value2&gt;);</code> <b>only ES5 types</b> , <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>isNumeric(&lt;value&gt;);</code> , <code>isBoolean(&lt;value&gt;);</code> , <code>isElement(&lt;value&gt;);</code> , <code>isDate(&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>isRegex(&lt;value&gt;);</code>	
	<b>ES6:</b> <code>isSymbol(&lt;value&gt;);</code> , <code>isMap(&lt;value&gt;);</code> , <code>isSet(&lt;value&gt;);</code> , <code>isWeakMap(&lt;value&gt;);</code> , <code>isWeakSet(&lt;value&gt;);</code> , <code>isIterator(&lt;value&gt;);</code> , <code>isIterable(&lt;value&gt;);</code>	

Collections
<b>ES5:</b> arrayUnion(<collection1>[,collectionN]);, arrayIntersection(<collection1>,<collection2>);, arrayDifference(<collection1>,<collection2>);, arraySymmetricDifference(<collection1>,<collection2>);, arrayKeys(<collection>);, arrayValues(<collection>);, arrayEntries(<collection>);, uniqueArray(<value>);, uniquePush(<array>,<value>);, arrayClear(<array>);, arrayRemove(<array>,<val>[,all]);, item(<collection>,<index>);, range(<start>,<end>[,step]);, toPairs(<collection1>,<collection2>);, min(<collection>);, minIndex(<collection>);, max(<collection>);, maxIndex(<collection>);
<b>ES6:</b> setUnion(<collection1>[,collectionN]);, setIntersection(<set1>,<set2>);, setDifference(<set1>,<set2>);, setSymmetricDifference(<set1>,<set2>);

Polyfills
<b>Array:</b> Array.prototype.values(); <b>ES6</b> , Array.from();, Array.of();, Array.prototype.fill();, Array.prototype.find();, Array.prototype.findIndex();, Array.prototype.includes();, Array.prototype.flat();, Array.prototype.flatMap();, Array.prototype.copyWithin();
<b>String:</b> 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();, String.fromCodePoint();, String.prototype.codePointAt();
<b>Object:</b> Object.create();, Object.assign();, Object.fromEntries();, Object.entries();, Object.values();, Object.is();, Object.getOwnPropertyDescriptors();
<b>DOM:</b> globalThis;, window.screenLeft;, window.screenTop;, NodeList.prototype.forEach();, ChildNode.after();, ChildNode.before();, ChildNode.remove();, ChildNode.replaceWith();, ParentNode.append();, ParentNode.prepend();, Element.prototype.matches();, Element.prototype.closest();, Element.prototype.toggleAttribute();, Element.prototype.getAttributeNames();
<b>Number:</b> Number.MIN_SAFE_INTEGER;, Number.MAX_SAFE_INTEGER;, Number.EPSILON;, Number.isInteger();, Number.isSafeInteger();, Number.isFinite();, Number.isNaN();, isNaN();, Number.parseInt();, Number.parseFloat();
<b>Math:</b> Math.acosh();, Math.asinh();, Math.atanh();, Math.cbrt();, Math.clz32();, Math.cosh();, Math.expm1();, Math.fround();, Math.hypot();, Math.imul();, Math.log1p();, Math.log10();, Math.log2();, Math.sign();, Math.sinh();, Math.tanh();, Math.trunc();

Celestra ES6E (with celestra-es6.js or celestra-es6.min.js)
<b>Polyfills:</b> GeneratorFunction()
<b>Functions:</b> forOf(<collection>,<callback>);, mapOf(<collection>,<callback>);, filterOf(<collection>,<callback>);, sliceOf(<collection>[,begin[,end]]);, takeOf(<collection>,<n>);, dropOf(<collection>,<n>);, iterRange([start[,step[,end]]]);, iterCycle(<iter>[,n]);, iterRepeat(<value>[,n]);, isGenerator(<value>);, iterTake(<collection>,<n>); <b>DEPRECATED</b> , iterDrop(<collection>,<n>); <b>DEPRECATED</b>