

Celestra cheatsheet – v2.4.0 – <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 (<selector>[,context]).each (callback) ;</code> <code>qs (<selector>[,context]).argument ;</code> <code>domReady (<function>) ;</code> <code>inherit (<subclass>,<superclass>) ;</code> <code>random ([max]) ;</code> <code>random (<min>,<max>) ;</code> <code>randomString ([length[,specChar]]) ;</code> <code>b64Encode (<string>) ;</code> <code>b64Decode (<string>) ;</code> <code>javaHash (<data>[,hexa]) ;</code> <code>importScript (<url>[,success]) ;</code> <code>importScripts (<scripts>) ;</code> <code>importScripts (<script1>[,scriptN]) ;</code> <code>importStyle (<href>[,success]) ;</code> <code>importStyles (<styles>) ;</code> <code>importStyles (<style1>[,styleN]) ;</code> <code>getUrlVar ([name]) ;</code> <code>getUrlVarFromString (<querystr>[,name]) ;</code> <code>obj2string (<object>) ;</code> <code>getType (<variable>[,type]) ;</code> <code>merge ([deep,]<target>,<source1>[,srcN]) ;</code> <code>extend ([deep,]<target>,<source1>[,srcN]) ;</code> <code>deepAssign (<target>,<source1>[,srcN]) ;</code> <code>getFullscreen () ;</code> <code>setFullscreenOn (<selector> or <element>) ;</code> <code>setFullscreenOff () ;</code> <code>getLocation (<success>[,error]) ;</code> <code>getDoNotTrack () ;</code> <code>form2array (<form>) ;</code> <code>form2string (<form>) ;</code> <code>strRemoveTags (<string>) ;</code> <code>strReverse (<string>) ;</code> <code>createFile (<filename>,<content>[,dType]) ;</code> <code>noConflict () ;</code> <code>version ;</code>	<code>domCreate (<type>[,properties[,innerHTML]]) ;</code> <code>domCreate (<element descriptive object>) ;</code> <code>domToElement (<htmlString>) ;</code> <code>domGetCSS (<element>,<property>) ;</code> <code>domSetCSS (<element>,<property>,<value>) ;</code> <code>domSetCSS (<element>,<properties>) ;</code> <code>domFadeIn (<element>[,duration[,display]]) ;</code> <code>domFadeOut (<element>[,duration]) ;</code> <code>domFadeToggle (<element>[,duration[,display]]) ;</code> <code>domShow (<element>[,display]) ;</code> <code>domHide (<element>) ;</code> <code>domToggle (<element>[,display]) ;</code> <code>domOn (<eventTarget>,<eventType>,<callback>) ;</code> <code>domOff (<eventTarget>,<eventType>,<callback>) ;</code> <code>domTrigger (<eventTarget>,<eventType>) ;</code> <code>domSiblings (<element>) ;</code>	<code>toFunction (<function>) ;</code> <code>bind (<function>,<context>) ;</code> <code>hasOwn (<object>,<property>) ;</code> <code>tap (<value>,<callback>) ;</code> <code>forEach (<collection>,<callback>) ;</code> <code>map (<collection>,<callback>) ;</code> <code>forIn (<object>,<callback>) ;</code> <code>mapIn (<object>,<callback>) ;</code> <code>forOf (<collection>,<cb>) ;</code> ES6 <code>mapOf (<collection>,<cb>) ;</code> ES6 <code>constant (<value>) ;</code> <code>identity (<value>) ;</code> <code>noop () ;</code> <code>T () ;</code> <code>F () ;</code>
AJAX and CORS		
	<code>getAjax (<url>,<format>,<success>[,error[,user,<password>]]) ;</code> <code>postAjax (<url>,<data>,<format>,<success>[,error[,user,<password>]]) ;</code> <code>getCors (<url>,<format>,<success>[,error[,user,<password>]]) ;</code> <code>postCors (<url>,<data>,<format>,<success>[,error[,user,<password>]]) ;</code> Ajax shorthands: <code>getJSON (<url>,<success>);</code> , <code>getText (<url>,<success>);</code>	
Cookie		
	<code>setCookie (<name>,<value>[,hours[,path[,domain[,secure[,HttpOnly]]]]) ;</code> <code>getCookie ([name]) ; and hasCookie (<name>) ;</code> <code>removeCookie (<name>[,path[,domain[,secure[,HttpOnly]]]]) ;</code>	
Type Checking		
	ES5: <code>isEqual (<value1>,<value2>);</code> only ES5 types , <code>isString (<value>);</code> , <code>isChar (<value>);</code> , <code>isNumber (<value>);</code> , <code>isInteger (<value>);</code> , <code>isFloat (<value>);</code> , <code>isNumeric (<value>);</code> , <code>isBoolean (<value>);</code> , <code>isElement (<value>);</code> , <code>isDate (<value>);</code> , <code>isObject (<value>);</code> <code>isEmptyObject (<value>);</code> , <code>isFunction (<value>);</code> , <code>isArray (<value>);</code> , <code>isEmptyArray (<value>);</code> , <code>isArraylike (<value>);</code> , <code>isNull (<value>);</code> , <code>isUndefined (<value>);</code> , <code>isNullOrUndefined (<value>);</code> , <code>isPrimitive (<value>);</code> , <code>isRegex (<value>);</code>	
	ES6: <code>isSymbol (<value>);</code> , <code>isMap (<value>);</code> , <code>isSet (<value>);</code> , <code>isWeakMap (<value>);</code> , <code>isWeakSet (<value>);</code> , <code>isIterator (<value>);</code>	

Collections

ES5: 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>);

ES6: setUnion(<collection1>[,collectionN]);, setIntersection(<set1>,<set2>);, setDifference(<set1>,<set2>);,
setSymmetricDifference(<set1>,<set2>);

Polyfills

Array: 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();

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();, String.fromCodePoint();, String.prototype.codePointAt();

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

DOM: 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();

Number: Number.MIN_SAFE_INTEGER;,, Number.MAX_SAFE_INTEGER;,, Number.EPSILON;,, Number.isInteger();, Number.isSafeInteger();,
Number.isFinite();, Number.isNaN();, isNaN();, Number.parseInt();, Number.parseFloat();

Math: 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();