

Celestra cheatsheet – v2.0.6 – <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(function (el, i) { el.arguments; });</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>getScript(<url>[,success]);</code> <code>getScripts(<scripts>);</code> <code>getStyle(<href>[,success]);</code> <code>getStyles(<styles>);</code> <code>getUrlVar([name]);</code> <code>getUrlVarFromString(<querystr>[,name]);</code> <code>obj2string(<object>);</code> <code>getType(<variable>[,type]);</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>constant(<value>);</code> <code>identity(<value>);</code> <code>noop();</code> <code>removeTags(<string>);</code> <code>createFile(<filename>,<content>[,dType]);</code> <code>fromEntries(<entries>);</code> <code>noConflict();</code> <code>celToWindow();</code> <code>version;</code> | <code>domCreate(<type>[,properties[,innerHTML]]);</code> <code>domCreate(<element descriptive object>);</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>toFunction(<function>);</code> <code>bind(<function>,<context>);</code> <code>forEach(<collection>,<callback>);</code> <code>each(<collection>,<callback>);</code> <code>map(<collection>,<callback>);</code> <code>forIn(<object>,<callback>);</code> <code>mapIn(<object>,<callback>);</code> <code>toArray(<object>);</code> <code>toObject(<array>);</code> <code>hasOwn(<object>,<property>);</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> <code>getJSON(<url>,<success>);</code> <code>getText(<url>,<success>);</code> | |
| Cookie | | |
| | <code>setCookie(<name>,<value>[,hours[,path[,domain[,secure[,HttpOnly]]]]]);</code> <code>getCookie([name]);</code> <code>hasCookie(<name>);</code> <code>removeCookie(<name>[,path[,domain[,secure[,HttpOnly]]]]);</code> | |
| Type checking | | |
| | <code>isString(<value>);, isChar(<value>);, isNumber(<value>);, isInteger(<value>);,</code> <code>isFloat(<value>);, isNumeric(<value>);, isBoolean(<value>);, isObject(<value>);</code> <code>isEmptyObject(<value>);, isFunction(<value>);, isArray(<value>);,</code> <code>isEmptyArray(<value>);, isArraylike(<value>);, isNull(<value>);,</code> <code>isUndefined(<value>);, isNullOrUndefined(<value>);, isPrimitive(<value>);,</code> <code>isRegexp(<value>);, isDate(<value>);, isElement(<value>);</code> ES6: <code>isSymbol(<value>);, isMap(<value>);, isSet(<value>);</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.getOwnPropertyDescriptors();, Object.is();

DOM: NodeList.prototype.forEach();, ChildNode.after();, ChildNode.before();, ChildNode.remove();, ChildNode.replaceWith();, ParentNode.append();, ParentNode.prepend();, Element.prototype.toggleAttribute();, Element.prototype.matches();, Element.prototype.closest();, 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();