Celestra cheatsheet – v2.1.0 – 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

`qsa(<selector>[,context]).each( function (el, i) { el.arguments; } );`
`qs(<selector>[,context]).argument;`
`domReady(<function>);`
`inherit(<subclass>,<superclass>);`
`random([max]);`
`random(<min>,<max>);`
`randomString([length[,specChar]]);`
`b64Encode(<string>);`
`b64Decode(<string>);`
`javaHash(<data>[,hexa]);`
`getScript(<url>[,success]);`
`getScripts(<scripts>);`
`getStyle(<href>[,success]);`
`getStyles(<styles>);`
`getUrlVar([name]);`
`getUrlVarFromString(<querystr>[,name]);`
`obj2string(<object>);`
`getType(<variable>[,type]);`
`extend([deep,]<target>,<source1>[,srcN]);`
`deepAssign(<target>,<source1>[,srcN]);`
`getFullscreen();`
`setFullscreenOn(<selector> or <element>);`
`setFullscreenOff();`
`getLocation(<success>[,error]);`
`getDoNotTrack();`
`form2array(<form>);`
`form2string(<form>);`
`constant(<value>);`
`identity(<value>);`
`noop();`
`removeTags(<string>);`
`createFile(<filename>,<content>[,dType]);`
`noConflict();`
`version;`
	`domCreate(<type>[,properties[,innerHTML]]);`
	`domCreate(<element descriptive object>);`
	`domGetCSS(<element>,<property>);`
	`domSetCSS(<element>,<property>,<value>);`
	`domSetCSS(<element>,<properties>);`
	`domFadeIn(<element>[, duration[,display]]);`
	`domFadeOut(<element>[,duration]);`
	`domFadeToggle(<element>[,duration[,display]]);`
	`domShow(<element>[,display]);`
	`domHide(<element>);`
	`domToggle(<element>[,display]);`
	`domOn(<eventTarget>,<eventType>,<callback>);`
	`domOff(<eventTarget>,<eventType>,<callback>);`
	`domTrigger(<eventTarget>,<eventType>);`
		`toFunction(<function>);`
		`bind(<function>,<context>);`
		`forEach(<collection>,<callback>);`
		`each(<collection>,<callback>);`
		`map(<collection>,<callback>);`
		`forIn(<object>,<callback>);`
		`mapIn(<object>,<callback>);`
		`toArray(<object>);`
		`toObject(<array>);`
		`hasOwn(<object>,<property>);`

AJAX and CORS

```
getAjax(<url>,<format>,<success>[,error[,user,<password>]]);
postAjax(<url>,<data>,<format>,<success>[,error[,user,<password>]]);
getCors(<url>,<format>,<success>[,error[,user,<password>]]);
postCors(<url>,<data>,<format>,<success>[,error[,user,<password>]]);
getJson(<url>,<success>);
getText(<url>,<success>);
```

Cookie

```
setCookie(<name>,<value>[,hours[,path[,domain[,secure[,HttpOnly]]]]]);
getCookie([name]);
hasCookie(<name>);
removeCookie(<name>[,path[,domain[,secure[,HttpOnly]]]]);
```

Type checking

```
isString(<value>);, isChar(<value>);, isNumber(<value>);, isInteger(<value>);,
isFloat(<value>);, isNumeric(<value>);, isBoolean(<value>);, isObject(<value>);
isEmptyObject(<value>);, isFunction(<value>);, isArray(<value>);,
isEmptyArray(<value>);, isArraylike(<value>);, isNull(<value>);,
isUndefined(<value>);, isNullOrUndefined(<value>);, isPrimitive(<value>);,
isRegexp(<value>);, isDate(<value>);, isElement(<value>);
```
ES6: `isSymbol(<value>);, isMap(<value>);, isSet(<value>);`

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: 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();

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();