Celestra cheatsheet – v2.7.2 – https://github.com/Serrin/Celestra/

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

Core API

```
qsa(<selector>[,context]).forEach(<fn>);
qs(<selector>[,context]).argument;
domReady(<function>);
inherit(<subclass>,<superclass>);
random(<min>,<max> or [max]);
randomString([length[,specChar]]);
b64Encode(<string>);
b64Decode(<string>);
javaHash(<data>[,hexa]);
importScript(<url>[,success]);
importScripts(<scripts>);
importScripts(<script1>[,scriptN]);
importStyle(<href>[,success]);
importStyles(<styles>);
importStyles(<style1>[,styleN]);
getUrlVar([name]);
getUrlVarFromString(<querystr>[,name]);
obj2string(<object>);
getType(<variable>[,type]);
extend([deep,]<target>,<source1>[,srcN]);
deepAssign(<target>,<source1>[,srcN]);
forIn(<object>,<callback>);
mapIn(<object>,<callback>);
getFullscreen();
setFullscreenOn(<selector> or <element>);
setFullscreenOff();
getLocation(<success>[,error]);
getDoNotTrack();
form2array(<form>);
form2string(<form>);
strRemoveTags(<string>);
strReverse(<string>);
createFile(<filename>,<content>[,dType]);
toFunction(<function>);
bind(<function>,<context>);
hasOwn(<object>,<property>);
constant(<value>);, identity(<value>);
noop(); and T(); and F();
noConflict(); and version;
```

DOM

```
domCreate(<type>[,properties[,innerHTML]]);
domCreate(<element descriptive object>);
domToElement(<htmlString>);
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(<el>);
domToggle(<element>[,display]);
domIsHidden(<element>);
domOn(<eventTarget>,<eventType>,<callback>);
domOff(<eventTarget>,<eventType>,<callback>);
domTrigger(<eventTarget>,<eventType>);
domSiblings(<element>);
```

ES6E (with celestra-es6.min.js)

```
itemOf(<collection>,<index>);
forOf(<collection>,<callback>);
mapOf(<collection>,<callback>);
filterOf(<collection>,<callback>);
sliceOf(<collec.>[,begin[,end]]);
takeOf(<collection>,<n>);
takeWhile(<collection>,<cb>);
dropOf(<collection>,<n>);
dropWhile(<collection>,<cb>);
iterRange([start[,step[,end]]]);
iterCycle(<iter>[,n]);
iterRepeat(<value>[,n]);
isGenerator(<value>);
```

Polyfills:

```
GeneratorFunction();
String.prototype.matchAll();
```

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>]]);
```

Ajax shorthands: `getJson(<url>,<success>);`, `getText(<url>,<success>);`

Cookie

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

Type Checking

ES5: `isEqual(<value1>,<value2>);` **only ES5 types** , `isString(<value>);`, `isChar(<value>);`, `isNumber(<value>);`, `isInteger(<value>);`, `isFloat(<value>);`, `isNumeric(<value>);`, `isBoolean(<value>);`, `isElement(<value>);`, `isDate(<value>);`, `isObject(<value>);`, `isEmptyObject(<value>);`, `isFunction(<value>);`, `isArray(<value>);`, `isEmptyArray(<value>);`, `isArraylike(<value>);`, `isNull(<value>);`, `isUndefined(<value>);`, `isNullOrUndefined(<value>);`, `isNil(<value>);`, `isPrimitive(<value>);`, `isRegexp(<value>);`

ES6: `isSymbol(<value>);`, `isMap(<value>);`, `isSet(<value>);`, `isWeakMap(<value>);`, `isWeakSet(<value>);`, `isIterator(<value>);`, `isIterable(<value>);`, `isBigInt(<v>);`

Collections
ES5: forEach(<collection>,<callback>);, map(<collection>,<callback>);, item(<collection>,<index>);, 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]);, arrayMerge([deep,]<target>,<source1>[,srcN]);, 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.prototype.values(); **ES6** , 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();
RegExp: RegExp.prototype.flags
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();