Celestra cheatsheet – v4.4.0 – https://github.com/Serrin/Celestra/

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

Core API	**DOM**	**Type checking**
signbit(<value>);	qsa(<selector>[,context]).forEach(<cb>);	isMap(<value>); *and* isWeakMap(<v>);
delay(<ms>).then(<callback>);	qs(<selector>[,context]);	isSet(<value>); *and* isWeakSet(<v>);
inherit(<subclass>,<superclass>);	domReady(<callback>);	isNumber(<v>); *and* isNumeric(<v>);
randomInt([max] *or* <min>,<max>);	domCreate(<type>[,properties[,innerHTML]]);	isFloat(<v>); *and* isBigInt(<v>);
randomFloat([max] *or* <min>,<max>);	domCreate(<element descriptive object>);	isString(<v>); *and* isChar(<v>);
randomBoolean();	domToElement(<htmlString>);	isDate(<v>); *and* isError(<v>);
randomString([length[,specChar]]);	domGetCSS(<element>[,property]);	isRegexp(<v>); *and* isSymbol(<v>);
b64Encode(<string>);, b64Decode(<str>);	domSetCSS(<element>,<property>,<value>);	isElement(<v>); *and* isObject(<v>);
javaHash(<data>[,hexa]);	domSetCSS(<element>,<properties>);	isNull(<value>);
getUrlVars([str=location.search]);	domFadeIn(<element>[,duration[,display]]);	isUndefined(<value>);
obj2string(<object>);	domFadeOut(<element>[,duration]);	isNullOrUndefined(<value>);
getType(<variable>[,type]);	domFadeToggle(<elem.>[,duration[,display]]);	isNil(<value>);
extend([deep,]<target>,<source1>[,srcN]);	domShow(<element>[,display]);	isFunction(<value>);
deepAssign(<target>,<source1>[,srcN]);	domHide(<element>);	isGeneratorFn(<value>);
sizeIn(<obj>); *and* forIn(<obj>,<cb>);	domToggle(<element>[,display]);	isAsyncFn(<value>);
filterIn(<object>,<callback>);	domIsHidden(<element>);	isDataView(<value>);
popIn(<object>,<property>);	domSiblings(<element>);	isBoolean(<value>);
strCapitalize(<string>);	domSiblingsPrev(<element>);	isArraylike(<value>);
strUpFirst(<str>);, strDownFirst(<str>);	domSiblingsLeft(<element>);	isTypedArray(<value>);
strHTMLRemoveTags(<string>);	domSiblingsNext(<element>);	isArrayBuffer(<value>);
strHTMLEscape(<string>);	domSiblingsRight(<element>);	isPrimitive(<value>);
strHTMLUnEscape(<string>);	domGetCSSVar(<name>);	isIterator(<value>);
strReverse(<string>);	domSetCSSVar(<name>,<value>);	isIterable(<value>);
strCodePoints(<string>);	importScript(<url>[,success]);	isPromise(<value>);
strFromCodePoints(<collection>);	importScripts(<scripts> *or* <script1>[,scN]);	isEmptyObject(<value>);
strAt(<string>,<index>);	importStyle(<href>[,success]);	isEmptyArray(<value>);
toFunction(<fn>);, bind(<fn>,<context>);	importStyles(<styles> *or* <style1>[,styleN]);	isEmptyMap(<value>);
constant(<value>); *and* identity(<value>);	setFullscreenOn(<selector> *or* <element>);	isEmptySet(<value>);
noop(); *and* T(); *and* F();	setFullscreenOff();	isEmptyIterator(<value>);
assertEq(<msg>,<v1>,<v2>[,strict=true]);	getFullscreen();	isSameObject(<object1>,<object2>);
assertNotEq(<m>,<v1>,<v2>[,strict=true]);	form2array(<form>); *and* form2string(<form>);	isSameArray(<array1>,<array2>);
assertTrue(<msg>,<value>);	getDoNotTrack();	isSameMap(<map1>,<map2>);
assertFalse(<msg>,<value>);	getLocation(<success>[,error]);	isSameSet(<set>,<set2>);
noConflict(); *and* VERSION;	createFile(<filename>,<content>[,dType]);	isSameIterator(<iter1>,<iter2>);

AJAX and CORS

ajax(<Options object>);, getJson(<url>,<success>);, getText(<url>,<success>);
Options object properties (* = default value): url: *string*, data: *string*, queryType: *"ajax"/"cors", type: *"get"/"post", success: *function*, error: *function*, format: *"text"/"json"/"xml", user: *string*, password: *string*

Collections

```
arrayMerge([deep,]<target>,<source1>[,srcN]);
arrayUnique(<collection>);
arrayAdd(<array>,<value>);
arrayClear(<array>);
arrayRemove(<array>,<value>[,all]);
arrayRange([start=0[,end=100[,step=1]]]);
arrayCycle(<collection>[,n]);
arrayRepeat(<value>[,n]);
iterRange([start=0[,step=1[,end]]]);
iterCycle(<iter>[,n]);
iterRepeat(<value>[,n]);

arrayUnion(<collection1>[,collectionN]);
arrayIntersection(<collection1>,<collection2>);
arrayDifference(<collection1>,<collection2>);
arraySymmetricDifference(<collection1>,<collection2>);
setUnion(<collection1>[,collectionN]);
setIntersection(<set1>,<set2>);
setDifference(<set1>,<set2>);
setSymmetricDifference(<set1>,<set2>);
isSuperset(<superCollection>,<subCollection>);

withOut(<collection>,<filterCollection>);
reduce(<collection>,<callback>[,initialvalue]);

take(<collection>[,n]);
takeWhile(<collection>,<callback>);
takeRight(<collection>[,n]);
takeRightWhile(<collection>,<callback>);
drop(<collection>[,n]);
dropWhile(<collection>,<callback>);
dropRight(<collection>[,n]);
dropRightWhile(<collection>,<callback>);
```

```
forEach(<collect.>,<callback>);
map(<collection>,<callback>);
enumerate(<collection>[,offset=0]);
entries(<collection>[,offset=0]);
size(<collection>);
every(<collection>,<callback>);
some(<collection>,<callback>);
none(<collection>,<callback>);
includes(<collection>,<value>);
find(<collection>,<callback>);
filter(<collection>,<callback>);
min(<collection>);
max(<collection>);
sort(<collection>[,numberSort]);
reverse(<collection>);
shuffle(<collection>);
partition(<collection>,<callback>);
groupBy(<collection>,<callback>);
zip(<collection1>[,collectionN]);
unzip(<collection>);

item(<collection>,<index>);
nth(<collection>,<index>);
first(<collection>);
head(<collection>);
last(<collection>);
initial(<collection>);
tail(<collection>);
slice(<collection>[,begin[,end]]);

flat(<collection>);
concat(<collection1>[,collectionN]);
join(<collection>[,separator=","]);
```

Polyfills

```
Array.prototype.at();
Array.prototype.flat();
Array.prototype.flatMap();

globalThis;

Object.fromEntries();
Object.hasOwn();

String.prototype.at();
String.prototype.matchAll();
String.prototype.padStart();
String.prototype.padEnd();
String.prototype.replaceAll();
String.prototype.trimStart();
String.prototype.trimLeft();
String.prototype.trimEnd();
String.prototype.trimRight();

TypedArray.prototype.at();
```

Non-standard polyfills

```
BigInt.prototype.toJSON();

window.AsyncFunction();
window.GeneratorFunction();
```

Cookie

```
getCookie([name]);, hasCookie(<name>);,
setCookie(<name>,<value>[,hours=8760[,path="/"[,domain[,secure[,SameSite="Lax"[,HttpOnly]]]]]]);, setCookie(<Optionsobj>);
removeCookie(<name>[,path="/"[,domain[,secure[,SameSite="Lax"[,HttpOnly]]]]]);, removeCookie(<Options object>);,
clearCookies([path="/"[,domain[,sec[,SameSite="Lax"[,HttpOnly]]]]]);, clearCookies(<Options object>);
```