Celestra cheatsheet – v3.0.0 – https://github.com/Serrin/Celestra/

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

DOM

```
qsa(<selector>[,context]).forEach(<fn>);
qs(<selector>[,context]).argument;
domReady(<function>);
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(<elem.>[,duration[,display]]);
domShow(<element>[,display]);
domHide(<element>);
domToggle(<element>[,display]);
domIsHidden(<element>);
domOn(<eventTarget>,<eventType>,<callback>);
domOff(<eventTarget>,<eventType>,<callback>);
domTrigger(<eventTarget>,<eventType>);
domSiblings(<element>);
importScript(<url>[,success]);
importScripts(<scripts> or  <script1>[,scN]);
importStyle(<href>[,success]);
importStyles(<styles> or <style1>[,styleN]);
getFullscreen();
setFullscreenOn(<selector> or <element>);
setFullscreenOff();
form2array(<form>);
form2string(<form>);
getDoNotTrack();
getLocation(<success>[,err]);
createFile(<filename>,<content>[,dType]);
```

Core API

```
inherit(<subclass>,<superclass>);
random(<min>,<max> or [max]);
randomString([length[,specChar]]);
b64Encode(<string>);
b64Decode(<string>);
javaHash(<data>[,hexa]);
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>);
strRemoveTags(<string>);
strReverse(<string>);
toFunction(<function>);
bind(<function>,<context>);
hasOwn(<object>,<property>);
constant(<value>);
identity(<value>);
noop();
T();
F();
noConflict();
VERSION;
```

Type checking

```
isEqual(<value1>,<value2>);  ES5
isString(<v>); and isChar(<v>);
isNumber(<v>); and isNumeric(<v>);
isInteger(<v>); and isFloat(<v>);
isBigInt(<value>);
isDate(<value>);
isBoolean(<value>);
isElement(<value>);
isObject(<value>);
isEmptyObject(<value>);
isFunction(<value>);
isArray(<value>);
isArraylike(<value>);
isEmptyArray(<value>);
isTypedArray(<value>);
isArrayBuffer(<value>);
isNull(<v>); and isUndefined(<v>);
isNullOrUndefined(<value>);
isNil(<value>);
isPrimitive(<value>);
isRegexp(<value>);
isSymbol(<value>);
isMap(<v>); and isWeakMap(<v>);
isSet(<v>); and isWeakSet(<v>);
isIterator(<value>);
isIterable(<value>);
isGenerator(<value>);
```

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]);, hasCookie(<name>);,
removeCookie(<name>[,path[,domain[,secure[,HttpOnly]]]]);, clearCookies([path[,domain[,secure[,HttpOnly]]]]);
```

Collections
isSuperset(<superset>,<subset>);, forEach(<collection>,<callback>);, map(<collection>,<cb>);, item(<collection>,<index>);, uniqueArray(<value>);, uniquePush(<array>,<value>);, arrayClear(<array>);, arrayRemove(<array>,<value>[,all]);, arrayMerge([deep,]<target>,<source1>[,srcN]);, zip(<collection1>[,collectionN]);, unzip(<collection>);, min(<collection>);, minIndex(<collection>);, max(<collection>);, maxIndex(<collection>);
setUnion(<collection1>[,collectionN]);, setIntersection(<set1>,<set2>);, setDifference(<set1>,<set2>);, setSymmetricDifference(<set1>,<set2>);, arrayUnion(<collection1>[,collectionN]);, arrayIntersection(<collection1>,<collection2>);, arrayDifference(<collection1>,<collection2>);, arraySymmetricDifference(<collection1>,<collection2>);
arrayRange(<start>,<end>[,step]);, arrayCycle(<collection>[,n]);, arrayRepeat(<value>[,n]);, iterRange([start[,step[,end]]]);, iterCycle(<iter>[,n]);, iterRepeat(<value>[,n]);
takeOf(<collection>[,n]);, takeWhile(<collection>,<callback>);, takeRight(<collec.>[,n]);, takeRightWhile(<collec.>,<cb>);, dropOf(<collection>[,n]);, dropWhile(<collection>,<callback>);, dropRight(<collec.>[,n]);, dropRightWhile(<collec.>,<cb>);
everyOf(<collection>,<callback>);, someOf(<collection>,<cb>);, noneOf(<collection>,<cb>);, itemOf(<collection>,<index>);, sizeOf(<collec.>);, firstOf(<collec.>);, lastOf(<collec.>);, sliceOf(<collec.>[,begin[,end]]);, filterOf(<collec.>,<cb>);, reverseOf(<collec.);, sortOf(<collec.>);, hasOf(<collec.>,<value>);, findOf(<collec.>,<callback>);, forOf(<collec.>,<cb>);, mapOf(<collection>,<callback>);, concatOf(<collection1>[,collectionN]);, reduceOf(<collection>,<callback>[,initialvalue]);

Polyfills
Array: Array.prototype.values();, 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();, String.prototype.matchAll();
Object: Object.create();, Object.assign();, Object.fromEntries();, Object.entries();, Object.values();, Object.is();, Object.getOwnPropertyDescriptors();
DOM: 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.sign();, Math.fround();, Math.hypot();, Math.imul();, Math.log1p();, Math.log10();, Math.log2();, Math.sinh();, Math.tanh();, Math.trunc();
Other: GeneratorFunction();, globalThis;, RegExp.prototype.flags;