JavaScript cheatsheet – v3.2.0 – https://github.com/Serrin/Celestra/

Web Storage api and JSON	element.dataset & data-* attributes	TypedArray
IE8+	- IE11 compatible	IE10+11 compatible
	- element data-* attributes	
localStorage:	- no methods and events	<pre>new <typedarray>(); ES2017</typedarray></pre>
localStorage.length;		<pre>new <typedarray>(length);</typedarray></pre>
<pre>localStorage.key(index);</pre>	camelcase:	<pre>new <typedarray>(typedArray);</typedarray></pre>
<pre>localStorage.getItem(key);</pre>	element.data-name	<pre>new <typedarray>(object);</typedarray></pre>
<pre>localStorage.setItem(key, data);</pre>	-> element.dataset.name	<pre>new <typedarray>(buffer[,byteOffset[,leng]]);</typedarray></pre>
<pre>localStorage.removeItem(key);</pre>	element.data-first-second	
<pre>localStorage.clear();</pre>	-> element.dataset.firstSecond	<pre>Int8Array();</pre>
		-128 to 127, 1 byte, int8 t
sessionStorage:	set:	<pre>Uint8Array();</pre>
sessionStorage.length;	<pre>element.dataset.name = "value";</pre>	0 to 255, 1 byte, uint8 t
<pre>sessionStorage.key(index);</pre>	<pre>element.dataset["name"] = "value";</pre>	<pre>Uint8ClampedArray(); - not in IE10-11</pre>
<pre>sessionStorage.getItem(key);</pre>	element.setAttribute("data-name",	0 to 255, 1 byte, uint8 t
<pre>sessionStorage.setItem(key, data);</pre>	"value");	_
<pre>sessionStorage.removeItem(key);</pre>	<pre>element["data-name"] = "value";</pre>	<pre>Int16Array();</pre>
<pre>sessionStorage.clear();</pre>		-32768 to 32767, 2 byte, int16 t
	get:	<pre>Uint16Array();</pre>
hasItem:	element.dataset.name;	0 to 65535, 2 byte, uint16 t
<pre>localStorage.getItem(key) !== null</pre>	<pre>element.dataset["name"];</pre>	_
<pre>sessionStorage.getItem(key) !== null</pre>	<pre>element.getAttribute("data-name");</pre>	<pre>Int32Array();</pre>
	<pre>element["data-name"];</pre>	-2147483648 to 2147483647, 4 byte, int32 t
setJSON:		<pre>Uint32Array();</pre>
localStorage.setItem(key,	remove:	0 to 4294967295, 4 byte, uint32 t
<pre>JSON.stringify(object));</pre>	<pre>element.removeAttribute("data-name");</pre>	_
sessionStorage.setItem(key,		<pre>BigInt64Array(); - not in IE10-11</pre>
<pre>JSON.stringify(object));</pre>	check:	-2**63 to 2**63-1, 8 byte, int64 t
	<pre>element.hasAttribute("data-name");</pre>	BigUint64Array(); - not in IE10-11
getJSON:		0 to 2**64-1, 8 byte, uint64 t
<pre>JSON.parse(localStorage.getItem(key));</pre>		
JSON.parse(sessionStorage.getItem(key)		<pre>Float32Array();</pre>
);		1.2x10-38 to 3.4x1038, 4 byte, float
		Float64Array();
		5.0x10-324 to 1.8x10308, 8 byte, double

```
element.classList
                                                                                            JSON
IE10+IE11 don't have support for classList on SVG or MathML | IE8+
elements.
                                                                Valid Data Types
element.classList.add(String[,String]);
                                                                - string
IE10+11: ves (except the multiple arguments)
                                                                - number
                                                                - object (containing valid JSON values)
element.classList.remove(String[,String]);
                                                                - arrav
IE10+11: yes (except the multiple arguments)
                                                                - boolean
- Removing a class that does not exist, does NOT throw an - date
                                                                - null
error.
element.classList.contains(String);
                                                                Invalid Data Types
                                                                - function
IE10+11: yes
                                                                - Symbol
element.classList.toggle(String[,force]);
                                                                - NaN, Infinity, undefined - will be "null"
IE10+11: yes (except the second argument)
                                                                - an object with method(s) (functions)
- When only one argument is present: Toggle class value; if - Map, Set, WeakMap, WeakSet - fix: convert to array
class exists then remove it and return false, if not, then add - BigInt - fixed in Celestra - BigInt.prototype.toJSON();
it and return true.
- When a second argument is present: If the second argument | JSON.stringify(value[,replacer[,space]]);
evaluates to true, add specified class value, and if it Convert a JavaScript object to a JSON string.
evaluates to false, remove it.
                                                                JSON.stringify( { a: 1, b: "2", c: true } );
element.classList.item(Number);
                                                                // -> "{\"a\":1,\"b\":\"2\",\"c\":true}"
IE10+11: yes
                                                                JSON.stringify([1, 2, 3, 4, 5]);
element.classList.length;
                                                                // -> "[1,2,3,4,5]"
IE10+11: ves
                                                                JSON.parse(text[,reviver]);
element.classList.replace(oldClass, newClass);
                                                                Parses a JSON string and returns a JavaScript object.
IE10+11: No and the method isn't compatible with the Safari and
mobile browsers too.
                                                                JSON.parse(JSON.stringify( {a: 1, b: "2", c: true} ));
                                                                // -> Object { a: 1, b: "2", c: true }
Remove all classes:
element.className = "";
                                                                JSON.parse(JSON.stringify([1, 2, 3, 4, 5]));
                                                                // \rightarrow Array(5) [ 1, 2, 3, 4, 5]
```

```
DOMParser
                                                                                        BigInt
                                                                let theBiggestInt = 9007199254740991n;
IE9: XML support
                                                                let alsoHuge = BigInt(9007199254740991);
IE10+IE11: XML, SVG and HTML support
                                                                let hugeString = BigInt("9007199254740991");
                                                                let hugeHex = BigInt("0x1fffffffffffff");
var parser = new DOMParser();
var doc = parser.parseFromString("sourceStr", "application/xml");
                                                               |let hugeBin =
Returns a Document, but not a SVGDocument nor a HTMLDocument.
                                                                1111111");
                                                                // all -> 9007199254740991n
var parser = new DOMParser();
var doc = parser.parseFromString(sourceStr, "image/svg+xml");
Returns a SVGDocument, which also is a Document.
                                                                typeof 1n;
                                                                           // -> "bigint"
                                                                typeof BigInt("1"); // -> "bigint"
var parser = new DOMParser();
var doc = parser.parseFromString(sourceStr, "text/html");
                                                                Operators: +, *, -, **, %, Bitwise operators (e.g.: >>>)
Returns a HTML document.
                                                                ! On
                                                                        // -> true
                                                                        // -> false
                                                                !1n
                   DOMParser sample function
                                                                4n / 2n / / -> 2n
                                                                5n / 2n // -> 2n, not 2.5n -> rounded
function parseHTML (str) {
 return Array.from(
   (new DOMParser())
                                                                1n < 2 // -> true
     .parseFromString(str, "text/html")
                                                               |2n > 1 // -> true
     .childNodes[0]
                                                               2n > 2 // -> false
     .childNodes[1]
                                                                2n >= 2 // -> true
     .childNodes
                                                               let mixed = [4n, 6, -12n, 10, 4, 0, 0n];
 );
                                                               mixed.sort(); // \rightarrow [-12n, 0, 0n, 4n, 4, 6, 10]
                                                               Methods
parseHTML(
 "<div>123</div>"
                                                                BigInt.asIntN();
 + "<div>456</div>"
                                                                  - BigInt value to a signed integer
 + "<div>7</div>"
                                                                BigInt.asUintN();
 + "<q>>8"
                                                                  - BigInt value to an unsigned integer
                                                                BigInt.prototype.toLocaleString();
// -> Array(4) [ div, div, div, p ]
                                                                BigInt.prototype.toString();
// Tested in IE11, Edge, Firefox and Chrome.
                                                                BigInt.prototype.valueOf();
```