# JavaScript cheatsheet – v2.0.8 – https://github.com/Serrin/Celestra/

| Web Storage api and JSON | element.dataset & data-* attributes | TypedArray |
|---|---|---|
| IE8 compatible<br><br>**localStorage:**<br>localStorage.length;<br>localStorage.key(index);<br>localStorage.getItem(key);<br>localStorage.setItem(key, data);<br>localStorage.removeItem(key);<br>localStorage.clear();<br><br>**sessionStorage:**<br>sessionStorage.length;<br>sessionStorage.key(index);<br>sessionStorage.getItem(key);<br>sessionStorage.setItem(key, data);<br>sessionStorage.removeItem(key);<br>sessionStorage.clear();<br><br>**hasItem:**<br>localStorage.getItem(key) !== null<br>sessionStorage.getItem(key) !== null<br><br>**setJSON:**<br>localStorage.setItem(key,<br>JSON.stringify(object));<br>sessionStorage.setItem(key,<br>JSON.stringify(object));<br><br>**getJSON:**<br>JSON.parse(localStorage.getItem(key));<br>JSON.parse(sessionStorage.getItem(key)<br>); | - IE11 compatible<br>- element data-* attributes<br>- no methods and events<br><br>**camelcase:**<br>element.data-name<br>  -> element.dataset.name<br>element.data-first-second<br>  -> element.dataset.firstSecond<br><br>**set:**<br>element.dataset.name = "value";<br>element.dataset["name"] = "value";<br>element.setAttribute("data-name",<br>"value");<br>element["data-name"] = "value";<br><br>**get:**<br>element.dataset.name;<br>element.dataset["name"];<br>element.getAttribute("data-name");<br>element["data-name"];<br><br>**remove:**<br>element.removeAttribute("data-name");<br><br>**check:**<br>element.hasAttribute("data-name"); | IE10+11 compatible<br><br>new TypedArray(); *ES2017*<br>new TypedArray(length);<br>new TypedArray(typedArray);<br>new TypedArray(object);<br>new TypedArray(buffer[,byteOffset[,length]]);<br><br>**Int8Array();**<br>-128 to 127, 1 byte, int8_t<br><br>**Uint8Array();**<br>0 to 255, 1 byte, uint8_t<br><br>**Uint8ClampedArray();**<br>0 to 255, 1 byte, uint8_t, not in IE10-11<br><br>**Int16Array();**<br>-32768 to 32767, 2 byte, int16_t<br><br>**Uint16Array();**<br>0 to 65535, 2 byte, uint16_t<br><br>**Int32Array();**<br>-2147483648 to 2147483647, 4 byte, int32_t<br><br>**Uint32Array();**<br>0 to 4294967295, 4 byte, uint32_t<br><br>**Float32Array();**<br>1.2x10-38 to 3.4x1038, 4 byte, float<br><br>**Float64Array();**<br>5.0x10-324 to 1.8x10308, 8 byte, double |

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