

0

0

3

2

5

JSON - JavaScript Object Notation

Objects represented as dictionaries of name:value pairs
In JSON itself names and entries must be "double-quoted".

Accessing Pair values is done in two ways:
`object.propertyName` OR
`object["propertyName"]`

```
let car =
{
  make: 'Tesla' ,
  model: 'Model 3',
  makeAndModel:
    function(){ return(this.make + ' ' + this.model); }
};
document.getElementById("a").innerHTML = 'Make and Model: ' +
    car.makeAndModel();
document.getElementById("b").innerHTML = JSON.stringify(car);
```

```
// output
Make and Model: Tesla Model 3
{"make":"Tesla","model":"Model 3"}
```

Note that functions are elided by stringify

`JSON.parse()` and `JSON.stringify()` are duals of each other
`obj = JSON.parse(string)` takes a string and returns a JS object.
`string = JSON.stringify(obj)` takes a JS object and returns string.

JSON text representation of JS objects is useful for:
datafiles: configuration, graphics files, graph files AND
serialization of JavaScript runtime states.

```
localStorage.setItem("name", "value");
localStorage.getItem("name") returns value
localStorage allows up to 10 Mbytes of data
Example:
myJSON = JSON.stringify(obj);
localStorage.setItem('myJSON', myJSON);
myString = localStorage.getItem('myJSON');
```

How to retrieve (examine) all `localStorage` items
without knowing the keys in advance.