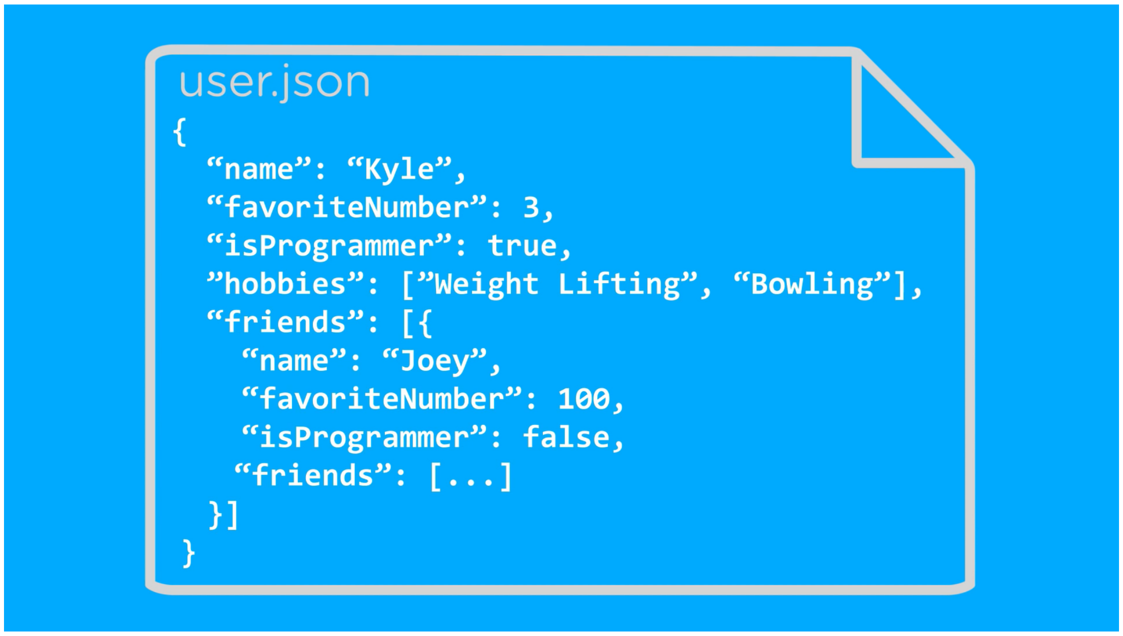
JSON

Notebook:	Animation Templates		
Created:	3/8/2020 3:02 AM	**Updated:**	3/15/2020 6:36 AM
Author:	Alan Tram		
URL:	https://www.w3schools.com/js/js_json_intro.asp		

Youtube Video https://www.youtube.com/watch?v=iiADhChRriM:

- Data Representation Format
- Commonly used for APIs and COnfigs
- Lightweight and Easy to Read/Write
- Integrates Easily With Most Languages


user.json

```
{
  “name”: “Kyle”,
  “favoriteNumber”: 3,
  “isProgrammer”: true,
  ”hobbies”: [”Weight Lifting”, “Bowling”],
  “friends”: [{
    “name”: “Joey”,
    “favoriteNumber”: 100,
    “isProgrammer”: false,
   “friends”: [...]
  }]
}
```

```
name = key | Kyle = value,
... (use commas to write additional key value pairs)
hobbies = key | [Weight Lifting" , "Bowling"] // array of values
...
//friends is an array of user objects. Nesting of arrays with objects
"friends" : [{
    "name":"Joey",
    "favoriteNumber":100,
    "isPRogrammer":false,
    "friends": [...]
```

JSON

- **J**ava**S**cript **O**bject **N**otation
- JSON is a syntax for storing and exchanging data
- JSON is text, written with JavaScript object notation

Exchanging Data

- When exchanging data between a browser and a server, the data can only be text
- JSON is text, and we can covnert any JS object into JSON, and send JSON to the server
- We can also convert any JSON received from the server into JS objects

```
<!DOCTYPE html>
<html>
<body>


<h2>Convert a JavaScript object into a JSON string, and send it to the server.
</h2>


<script>
var myObj = { name: "John", age: 31, city: "New York" };
var myJSON = JSON.stringify(myObj);
window.location = "demo_json.php?x=" + myJSON;
</script>


</body>
</html>
```

```
Output:

demo_json.php:
John from New York is 31
```

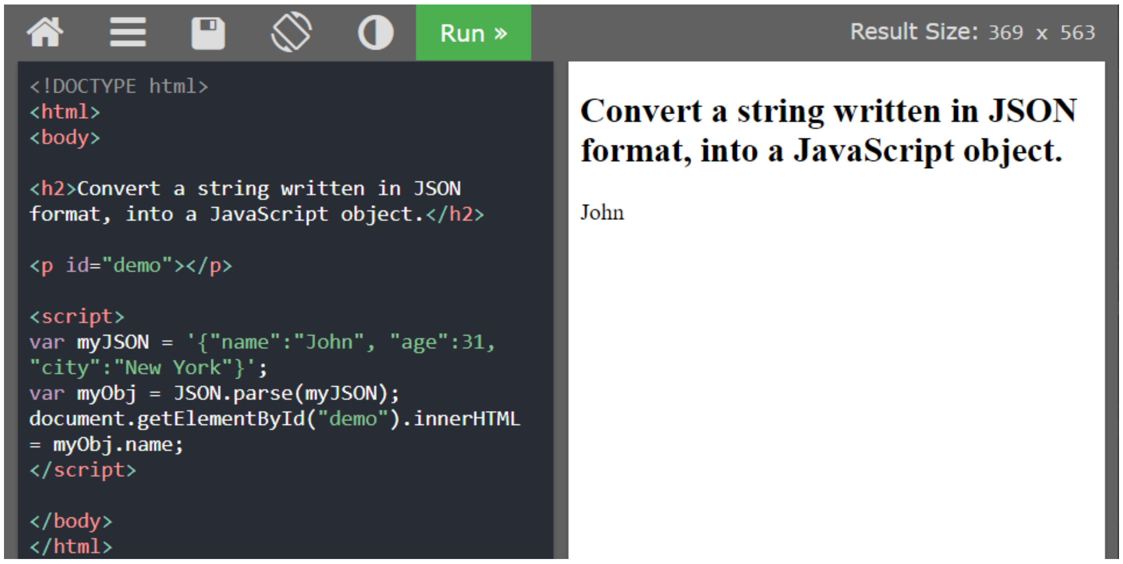
stringify() method converts a **JS object** or value to a **JSON string,** optionally replacing values if a replacer func is specified or optionally including only the specified properties if a replacer array is specified

Receiving Data

If you receive data in JSON format, you can convert it into JS object:

```
var myJSON = '{"name":"John" , "age":31, "city":"New York"}';

var myObj = JSON.parse(myJSON);
document.getElementById("demo").innerHTML = myObj.name;
```

output:



```
<!DOCTYPE html>
<html>
<body>

<h2>Convert a string written in JSON
format, into a JavaScript object.</h2>

<p id="demo"></p>

<script>
var myJSON = '{"name":"John", "age":31,
"city":"New York"}';
var myObj = JSON.parse(myJSON);
document.getElementById("demo").innerHTML
= myObj.name;
</script>

</body>
</html>
```

Convert a string written in JSON format, into a JavaScript object.

John

JSON.parse() method parses a JSON string, constructing the JS value or object described by the string.

Storing Data

When storing data, the data has to **be a certain format**, and regardless of where you choose to store it, text is always one of the legal formats.

JSON makes it possible to store JS objects as text

```
// Storing data:
myObj = {name: "John", age: 31, city: "New York"};
myJSON = JSON.stringify(myObj);
localStorage.setItem("testJSON", myJSON);


// Retrieving data:
text = localStorage.getItem("testJSON");
obj = JSON.parse(text);
document.getElementById("demo").innerHTML = obj.name;
```

JSON Data - A Name and a Value

JSON data is written as name/value pairs.

```
"name":"John"
// JSON names require double quotes
```

JSON - Evaluates to JavaScript Obj

The JSON format is almost identical to JS objects

In JSON, *keys* must be strings

```
var person = { name: "John", age: 31, city: "New York"};
```

```
//access JS object

person.name; //returns John

OR

person["name"]; //returns John
```

JavaScript Arrays as JSON

```
//JSON Objects
{
"employee: { "name":"John", "age":30, "city":"New York" }
}

//JSON array
{
"employees":["John","Anna","Peter"]
}
```

JSON.parse()

- A common use of JSON is to exchange data to/from a web server
- When receiving data from a web server, the data is always a string
- Parse the data with JSON.parse(), and the data becomes a **JS object**

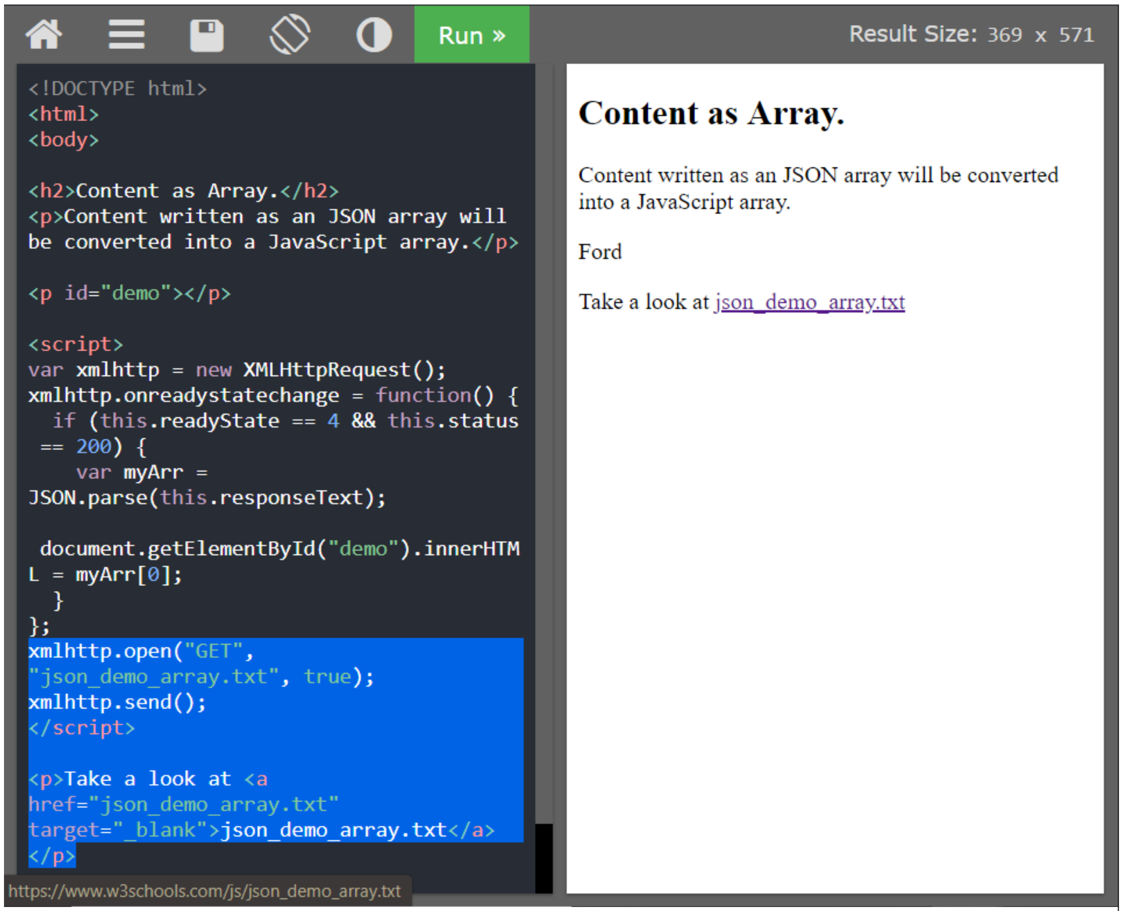
```
'{"name":"John", "age":30, "city":"New York"}'

// Use the JavaScript function JSON.parse() to convert text into a JS object

var obj = JSON.parse('{"name":"John", "age":30,"city":"New York"}');
```

Array as JSON

When using the **JSON.parse()** on a JSON derived from an array, the method will return a JavaScript array, instead of a JavaScript object

```
var xmlhttp = new XMLHttpRequest();
xmlhttp.onreadystatechange = function() {
  if (this.readyState == 4 && this.status == 200) {
    var myArr = JSON.parse(this.responseText);
    document.getElementById("demo").innerHTML = myArr[0];
  }
};
xmlhttp.open("GET", "json_demo_array.txt", true);
xmlhttp.send();
```

output:



```
<!DOCTYPE html>
<html>
<body>

<h2>Content as Array.</h2>
<p>Content written as an JSON array will
be converted into a JavaScript array.</p>

<p id="demo"></p>

<script>
var xmlhttp = new XMLHttpRequest();
xmlhttp.onreadystatechange = function() {
  if (this.readyState == 4 && this.status
 == 200) {
    var myArr =
JSON.parse(this.responseText);

 document.getElementById("demo").innerHTM
L = myArr[0];
  }
};
xmlhttp.open("GET",
"json_demo_array.txt", true);
xmlhttp.send();
</script>

<p>Take a look at <a
href="json_demo_array.txt"
target="_blank">json_demo_array.txt</a>
</p>
```

https://www.w3schools.com/js/json_demo_array.txt

Content as Array.

Content written as an JSON array will be converted into a JavaScript array.

Ford

Take a look at json_demo_array.txt

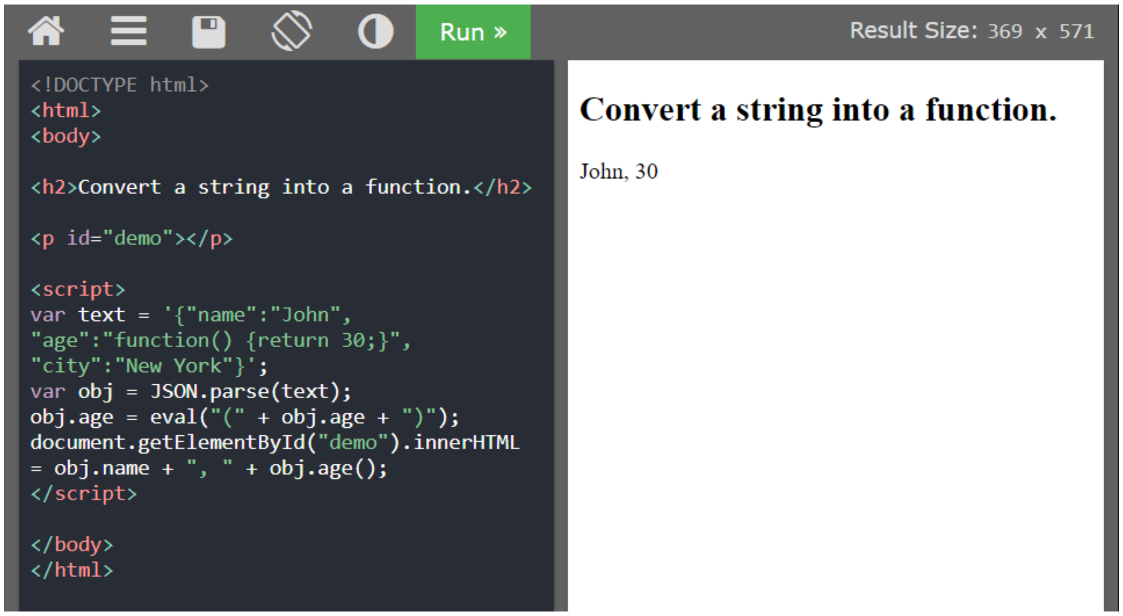
```
[ "Ford", "BMW", "Audi", "Fiat" ]
```

Parsing Functions

- Functions are not allowed in JSON.
- If you need to include a function, write it as a string
- Convert it back into a function later:

```
var text = '{"name":"John", "age":function () {return 30;}", "city":"New York"}';

var obj = JSON.parse(text);
obj.age = eval("(" + obj.age + ")");

document.getElementById("demo).innerHTML = obj.name + ", " + obj.age();
```

Output:



```
<!DOCTYPE html>
<html>
<body>

<h2>Convert a string into a function.</h2>

<p id="demo"></p>

<script>
var text = '{"name":"John",
"age":"function() {return 30;}",
"city":"New York"}';
var obj = JSON.parse(text);
obj.age = eval("(" + obj.age + ")");
document.getElementById("demo").innerHTML
= obj.name + ", " + obj.age();
</script>

</body>
</html>
```

Convert a string into a function.

John, 30

JSON.stringify()

- Data to a web server must be as a string

```
var obj = { name: "John" , age: 30 ,city: "New York"};

//convert to string

var myJSON = JSON.stringify(obj);

// myJSON is now the obj as string. Send to server

document.getElementById("demo").innerHTML = myJSON;

Output:

{"name":"John","age":30,"city":"New York"}
```

```
//Stringify a JS Array

var arr = [ "John", "Peter", "Sally", "Jane" ];

var myJSON = JSON.stringify(arr);

document.getElementById("demo").innerHTML = myJSON;

output:

["John","Peter","Sally","Jane"]
```

JSON Objects

- JSON objects are surrounded by curly braces {}
- JSON objects are written in key/value pairs
- Keys must be strings, and values must be a valid data type

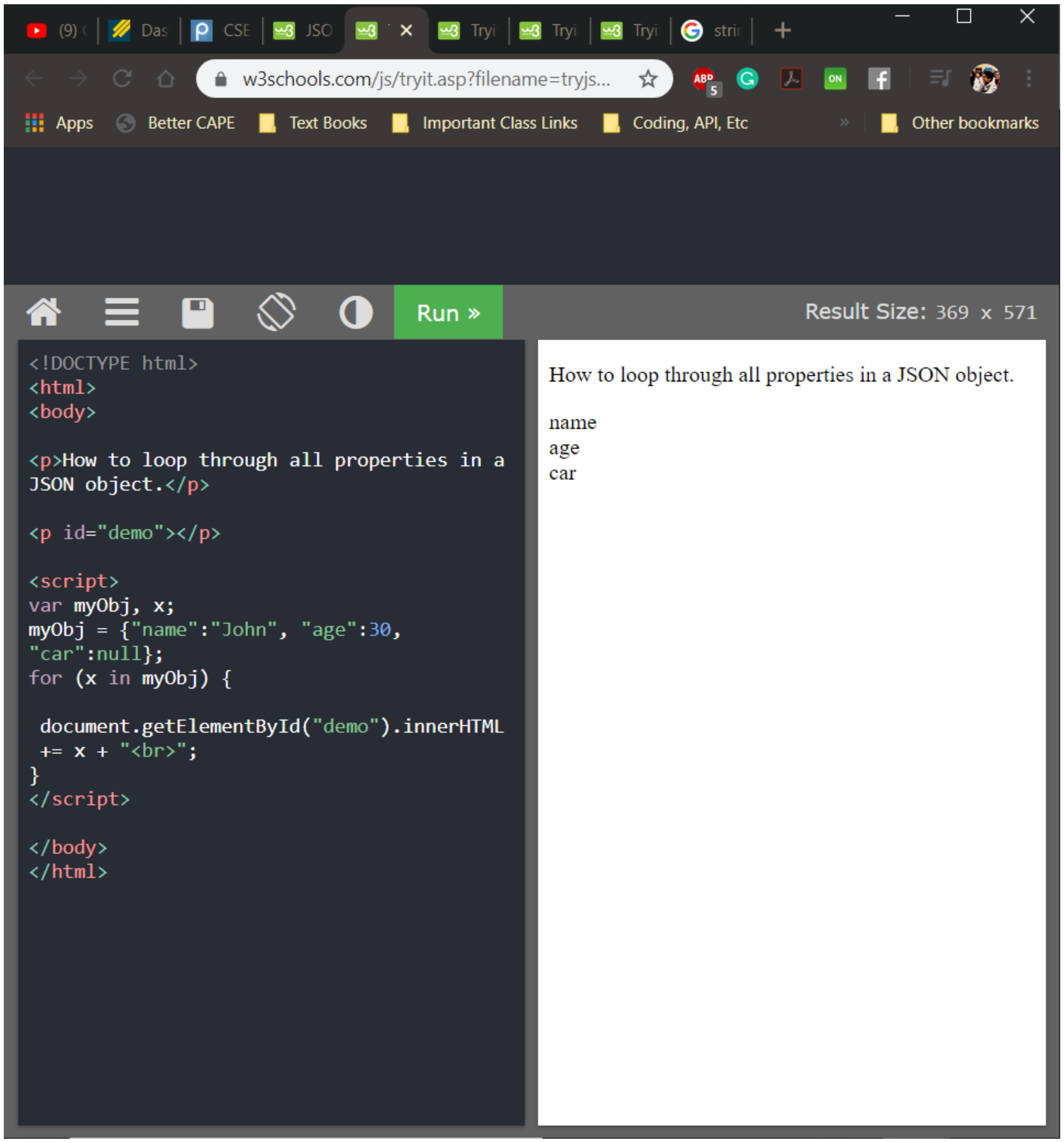
- Keys and values are separated by a colon
- Each key/value pair is separated by a comma

```
//Accessing Object Values

myObj = {"name":"John","age":30,"car":null};
x = myObj.name;

//Can also access by bracket)[]) notation

x = myObj["name"];
```

Looping an Object

```
//You can loop through object properties via for loop

myObj = {"name":"John", "age":30, "car":null};
for(x in myObj) {
    document.getElementById("demo").innerHTML += x;
}
```

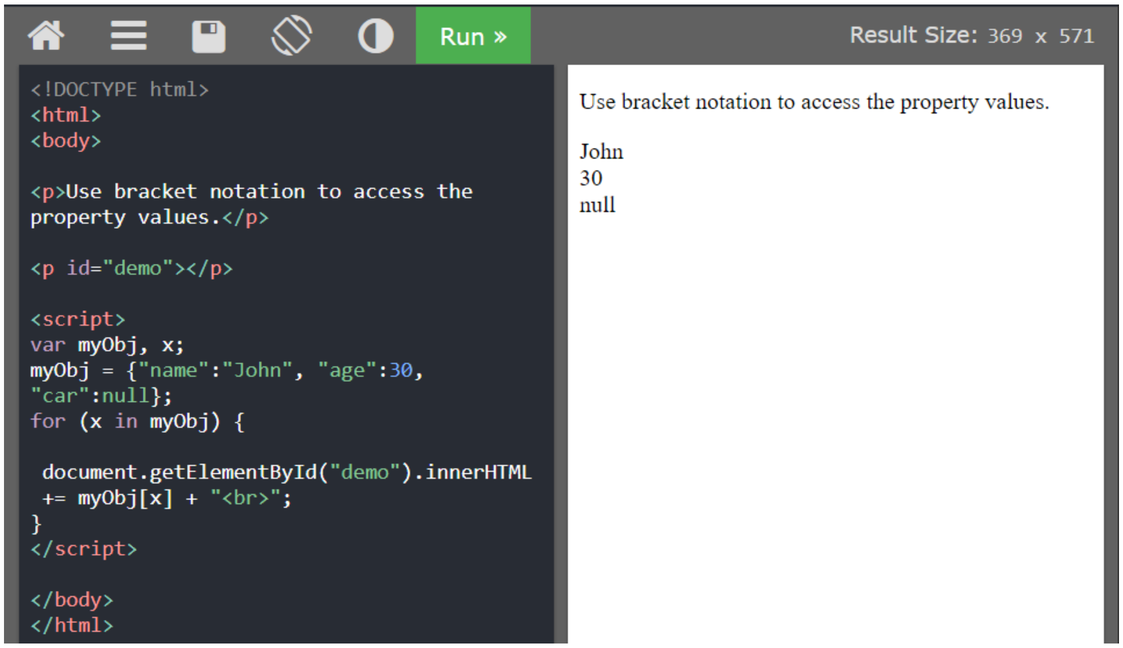
Output:



```
<!DOCTYPE html>
<html>
<body>

<p>How to loop through all properties in a
JSON object.</p>

<p id="demo"></p>

<script>
var myObj, x;
myObj = {"name":"John", "age":30,
"car":null};
for (x in myObj) {

 document.getElementById("demo").innerHTML
 += x + "<br>";
}
</script>

</body>
</html>
```

```
How to loop through all properties in a JSON object.

name
age
car
```

```
//In a for-in loop, use the bracket notation to access the property values:


for (x in myObj) {
  document.getElementById("demo").innerHTML += myObj[x];
}
```



```
<!DOCTYPE html>
<html>
<body>

<p>Use bracket notation to access the
property values.</p>

<p id="demo"></p>

<script>
var myObj, x;
myObj = {"name":"John", "age":30,
"car":null};
for (x in myObj) {
 document.getElementById("demo").innerHTML
 += myObj[x] + "<br>";
}
</script>

</body>
</html>
```

```
Use bracket notation to access the property values.

John
30
null
```

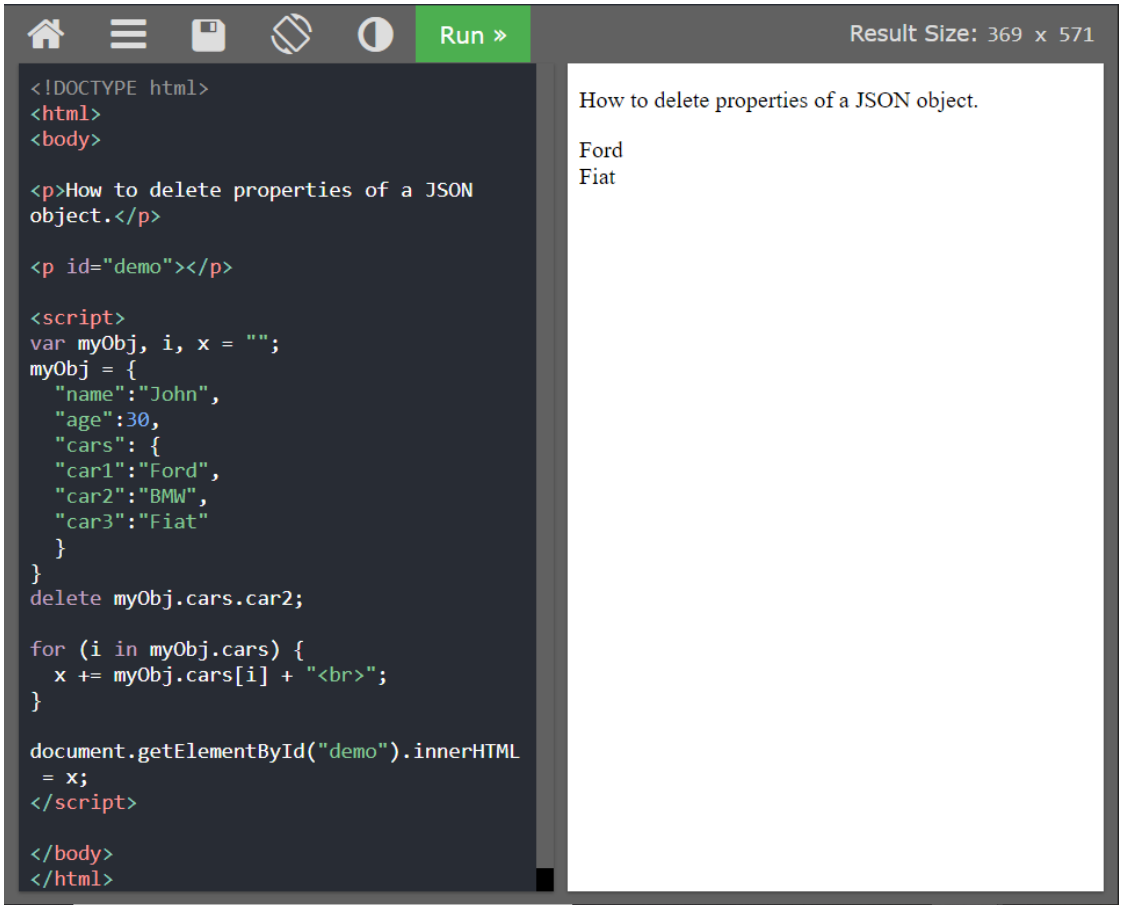
Nested JSON Objects

Values in a JSON object can be another JSON object

```
//Example Nested JSON Object

myObj = {
  "name":"John",
  "age":30,
  "cars": {
    "car1":"Ford",
    "car2":"BMW",
    "car3":"Fiat"
  }
}

/* You can access nested JSON objects by using the dot or *bracket notation
*/

x = myObj.cars.car2;
//or:
x = myObj.cars["car2"];
```

Delete Object Properties

```
delete myObj.cars.car2;

//delete BMW from nested JSON object
```



```
<!DOCTYPE html>
<html>
<body>

<p>How to delete properties of a JSON
object.</p>

<p id="demo"></p>

<script>
var myObj, i, x = "";
myObj = {
  "name":"John",
  "age":30,
  "cars": {
  "car1":"Ford",
  "car2":"BMW",
  "car3":"Fiat"
  }
}
delete myObj.cars.car2;

for (i in myObj.cars) {
  x += myObj.cars[i] + "<br>";
}

document.getElementById("demo").innerHTML
 = x;
</script>

</body>
</html>
```

```
How to delete properties of a JSON object.

Ford
Fiat
```

Arrays as JSON Objects

Arrays can be values of an object property:

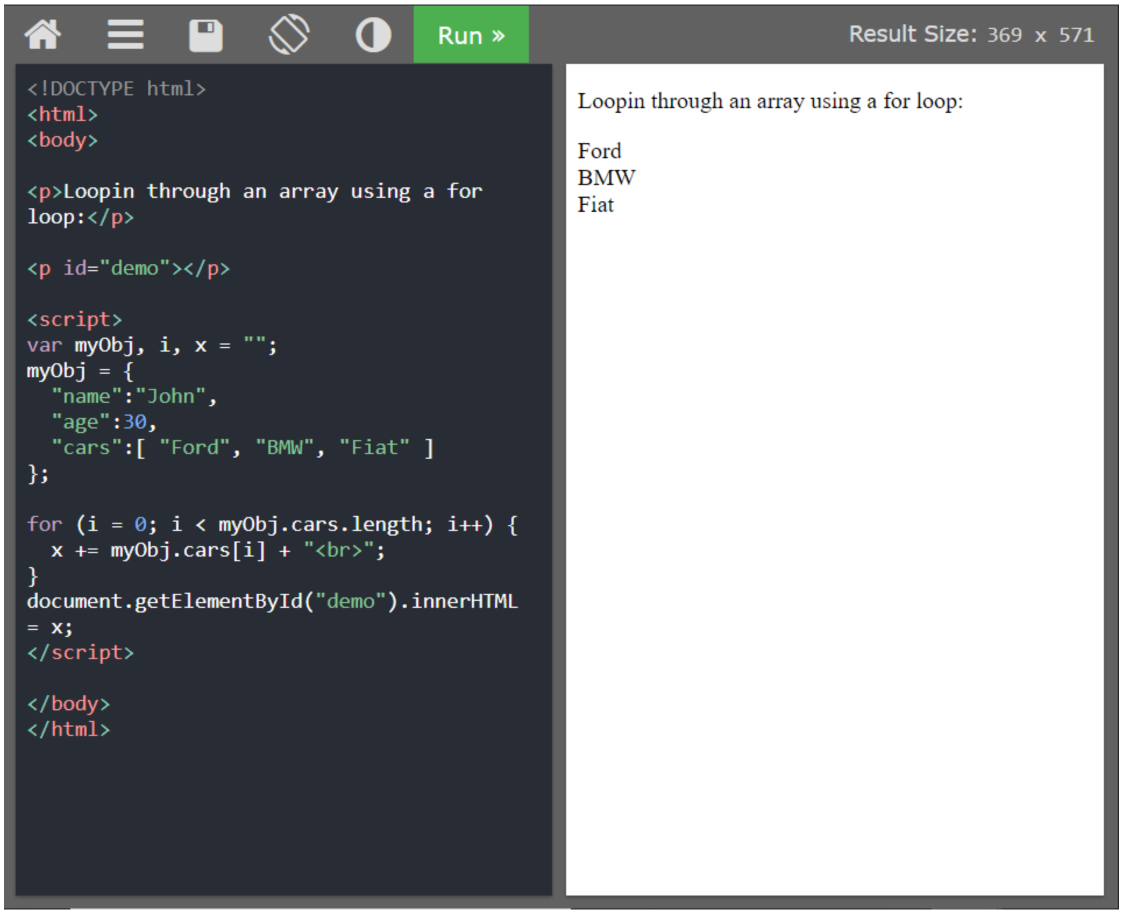
```
{
"name":"John",
"age":30,
"cars":[ "Ford", "BMW", "Fiat" ]
}
```

Accessing Array Values

```
x = myObj.cars[0];
```

Looping Through an Array

```
//You can access array values by using a for-in loop:

for (i in myObj.cars) {
  x += myObj.cars[i];
}

// for loop

for (i = 0; i < myObj.cars.length; i++) {
  x += myObj.cars[i];
}
```



```
<!DOCTYPE html>
<html>
<body>

<p>Loopin through an array using a for
loop:</p>

<p id="demo"></p>

<script>
var myObj, i, x = "";
myObj = {
  "name":"John",
  "age":30,
  "cars":[ "Ford", "BMW", "Fiat" ]
};

for (i = 0; i < myObj.cars.length; i++) {
  x += myObj.cars[i] + "<br>";
}
document.getElementById("demo").innerHTML
= x;
</script>

</body>
</html>
```

```
Loopin through an array using a for loop:

Ford
BMW
Fiat
```

Looping Through an Array

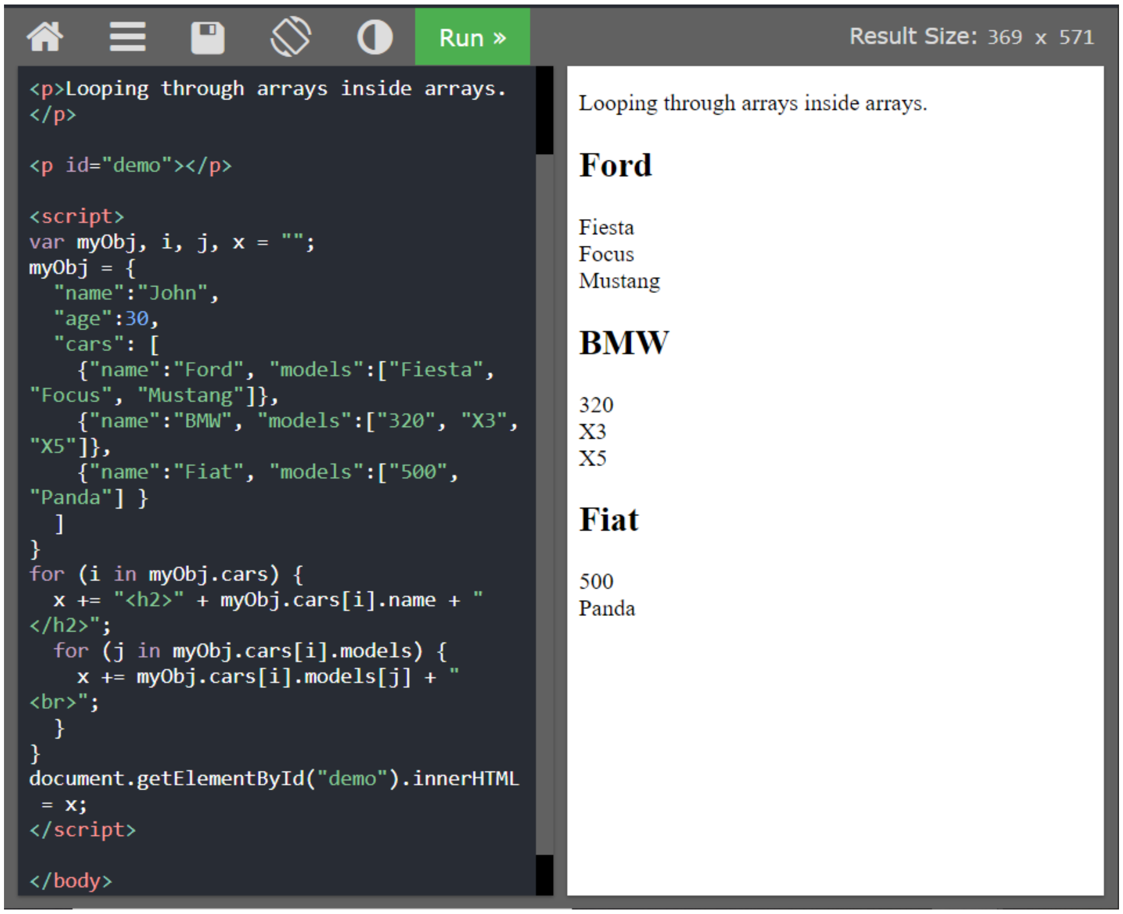
Values in an array can also be another array, or even another JSON object:

```
myObj = {
  "name":"John",
  "age":30,
  "cars": [
    { "name":"Ford", "models":[ "Fiesta", "Focus", "Mustang" ] },
    { "name":"BMW", "models":[ "320", "X3", "X5" ] },
    { "name":"Fiat", "models":[ "500", "Panda" ] }
  ]
}
```

To access arrays inside arrays, use a for-in loop for each array:

```
for (i in myObj.cars) {
  x += "<h1>" + myObj.cars[i].name + "</h1>";
  for (j in myObj.cars[i].models) {
    x += myObj.cars[i].models[j];
  }
}
```

Output:



```
<p>Looping through arrays inside arrays.
</p>

<p id="demo"></p>

<script>
var myObj, i, j, x = "";
myObj = {
  "name":"John",
  "age":30,
  "cars": [
    {"name":"Ford", "models":["Fiesta",
"Focus", "Mustang"]},
    {"name":"BMW", "models":["320", "X3",
"X5"]},
    {"name":"Fiat", "models":["500",
"Panda"] }
  ]
}
for (i in myObj.cars) {
  x += "<h2>" + myObj.cars[i].name + "
</h2>";
  for (j in myObj.cars[i].models) {
    x += myObj.cars[i].models[j] + "
<br>";
  }
}
document.getElementById("demo").innerHTML
 = x;
</script>

</body>
```

```
Looping through arrays inside arrays.

Ford

Fiesta
Focus
Mustang

BMW

320
X3
X5

Fiat

500
Panda
```