JSON

JavaScript Object Notation

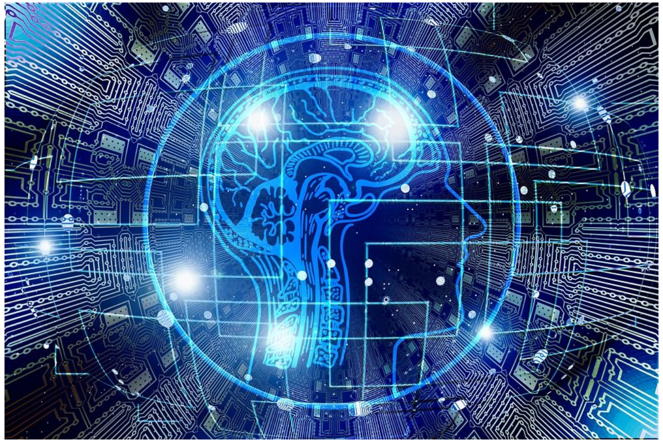
What is JSON

- Extended from the JavaScript
- Media type is application/json
- Extension is .json

Always starts and ends with curly brackets { }

Name and value is separated by a colon :

More than one pair is separated by comma ,



What is JSON

- JSON is a language-independent data format
- It derives from JavaScript
- JSON JavaScript Object Notation

It is the most common data format used for asynchronous browser/server communication, largely replacing XML which is used by AJAX

Basics of JSON

key/name value pairs

{ "name" : "value" }

Objects are comma separated

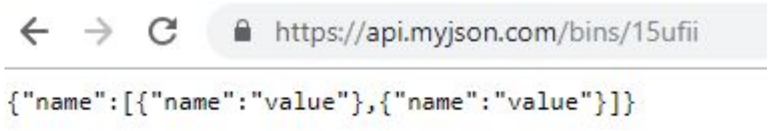
{ "name1" : "value" , "name2" : "value", "name3" : "value"}

Arrays have square brackets with values separated by comma

{ "name" : [{ "name" : "value" }, { "name" : "value" }] }

JSON lint makes more readable

```
{        "name": "value"      }


{        "name1": "value",
         "name2": "value",
         "name3": "value"    }


{        "name": [{
                  "name": "value"
         }, {
                  "name": "value"
         }]       }
```



```
{"name":[{"name":"value"},{"name":"value"}]}
```

Resources:
http://myjson.com - Storage of JSON

https://jsonlint.com - Validation of JSON

https://randomuser.me - JSON Data API

https://randomuser.me/api/?results=10

Try It

Open up your console.

Create an object

Output the object

Try in Lint

```
var myObj = { "firstName" : "Mike" ,"lastName" : "Smith" , "age": 30  };        console.log(myObj);
```

Data Structures

collection of name/value pairs : Think Object format

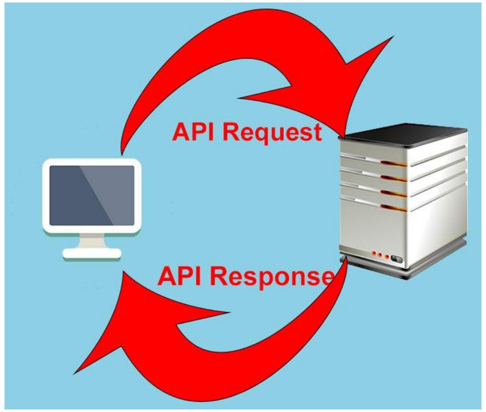
ordered list of values : Think Array format

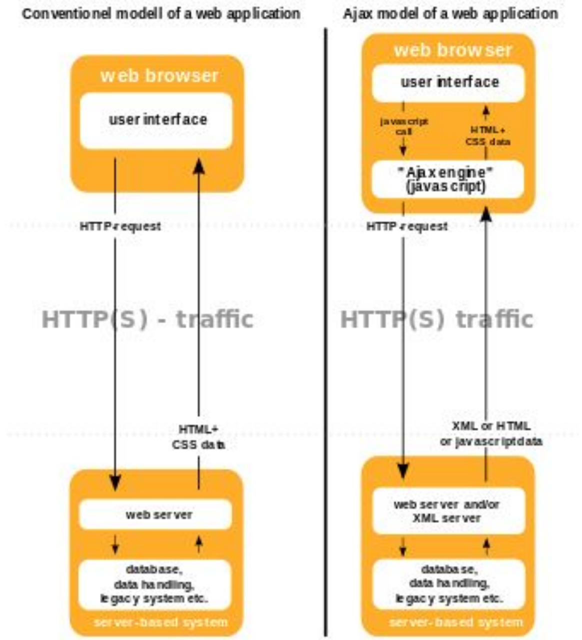
Structured Data

As many levels of lists as needed to organize the data.

APIS communication between software components

APIs are made up of requests and responses. **GET the DATA!!!!**




Conventionel modell of a web application
web browser
user interface
HTTP request
HTTP(S) - traffic
HTML+ CSS data
web server
database, data handling, legacy system etc.
server-based system
Ajax model of a web application
web browser
user interface
javascript call
HTML+ CSS data
"Ajax engine" (javascript)
HTTP request
HTTP(S) traffic
XML or HTML or javascript data
web server and/or XML server
database, data handling, legacy system etc.
server-based system

Data Types in JSON value can be any of these

Number - double- precision floating-point can be digits, positive or negative, decimal fractions, exponents… {"name":10}

String - double-quoted Unicode with backslash escaping {"name":"Hello world"}

Boolean - true or false {"name":true}

Array - ordered sequence of values uses square brackets. Values are each separated by a comma. Indexing starts with 0.   {"name": [{"name1": 1}, "hello", "world"]}

Object - unordered collection with key:value pairs. Wrapped with curly brackets {}. Colons to separate key and name. Keys should be strings and have unique names. {"name": {"name1": 1,"name2": 1}}

Null - just empty {"name": null}

How to create an Object

JSON is an object which can be used to describe something. Two items with one value described.

```
{
  "car1": "black",
  "car2": "blue"
}
```

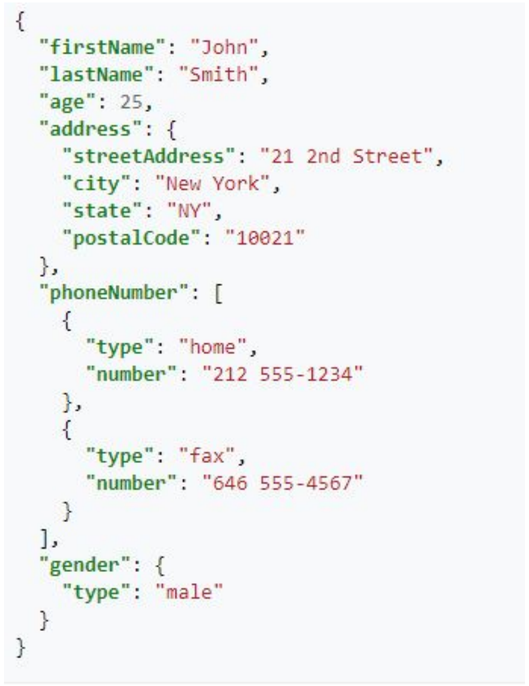
Tools:

https://jsonlint.com/

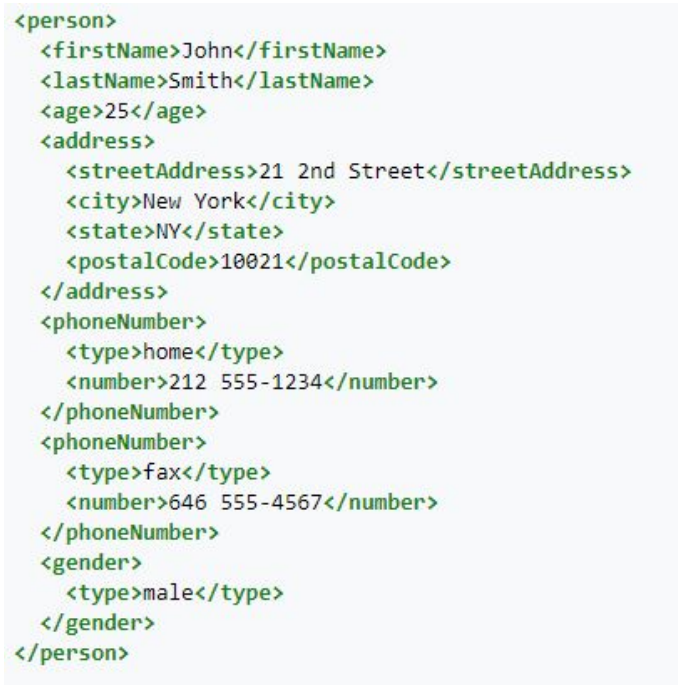
https://jsonschema.net/

JSON vs XML vs YAML

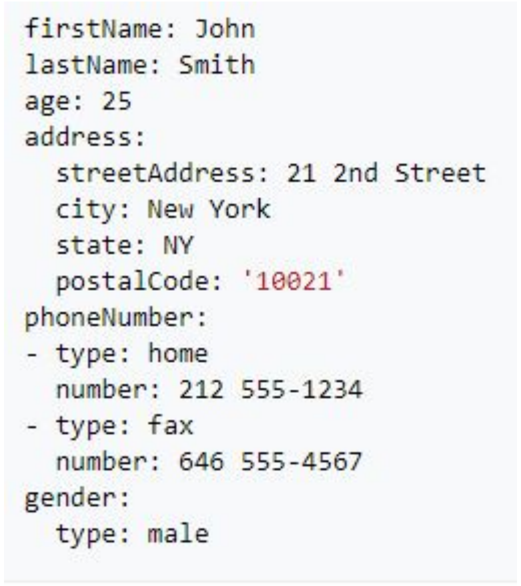
JSON and XML are human readable formats JSON is faster to write. XML has not arrays. JSON much easier to parse in JavaScript



```json
{
  "firstName": "John",
  "lastName": "Smith",
  "age": 25,
  "address": {
    "streetAddress": "21 2nd Street",
    "city": "New York",
    "state": "NY",
    "postalCode": "10021"
  },
  "phoneNumber": [
    {
      "type": "home",
      "number": "212 555-1234"
    },
    {
      "type": "fax",
      "number": "646 555-4567"
    }
  ],
  "gender": {
    "type": "male"
  }
}
```



```xml
<person>
  <firstName>John</firstName>
  <lastName>Smith</lastName>
  <age>25</age>
  <address>
    <streetAddress>21 2nd Street</streetAddress>
    <city>New York</city>
    <state>NY</state>
    <postalCode>10021</postalCode>
  </address>
  <phoneNumber>
    <type>home</type>
    <number>212 555-1234</number>
  </phoneNumber>
  <phoneNumber>
    <type>fax</type>
    <number>646 555-4567</number>
  </phoneNumber>
  <gender>
    <type>male</type>
  </gender>
</person>
```



```yaml
firstName: John
lastName: Smith
age: 25
address:
  streetAddress: 21 2nd Street
  city: New York
  state: NY
  postalCode: '10021'
phoneNumber:
- type: home
  number: 212 555-1234
- type: fax
  number: 646 555-4567
gender:
  type: male
```

Difference: JSON & JavaScript Object

JSON all *keys* must be quoted, object literals it is not necessary:

```
{ "foo": "bar" }
```

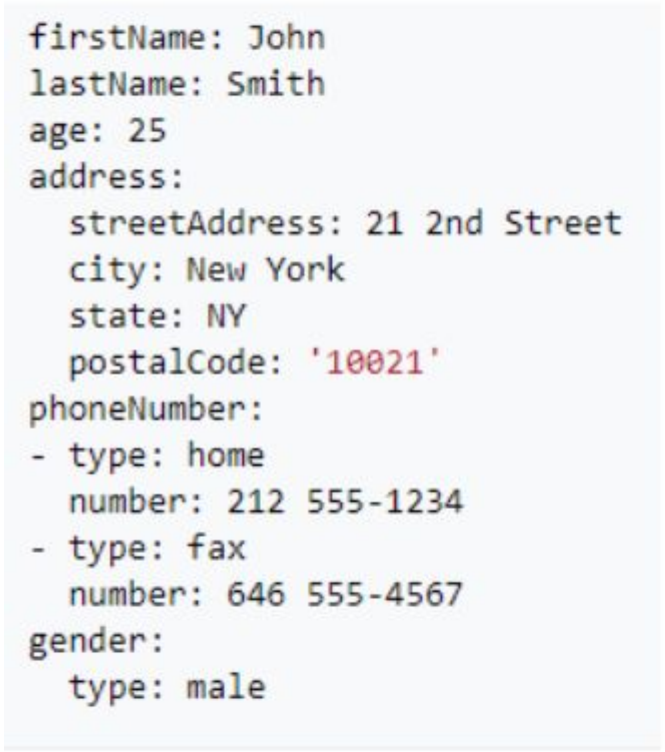
```
var o = { foo: "bar" };
```

JSON has double quotes while JavaScript can use single or doubles

JavaScript can include functions which is not available in JSON.

Try It

Create an object from this data.



```yaml
firstName: John
lastName: Smith
age: 25
address:
  streetAddress: 21 2nd Street
  city: New York
  state: NY
  postalCode: '10021'
phoneNumber:
- type: home
  number: 212 555-1234
- type: fax
  number: 646 555-4567
gender:
  type: male
```

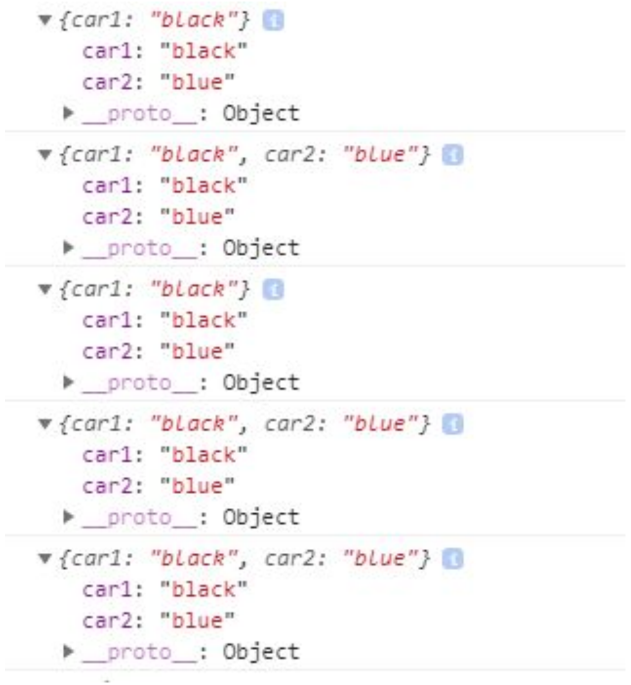
More details needed!!!

```
var myObj = {
"firstName" : "Mike" ,
"lastName" : "Smith" ,
"age": 30
};
console.log(myObj);
```

Objects in JavaScript

Try this in the console

```
var myJSON = {};
myJSON.car1 = "black"
console.log(myJSON)
myJSON.car2 = "blue"
console.log(myJSON)

var myJSON = {};
myJSON["car1"] = "black"
console.log(myJSON)
myJSON["car2"] = "blue"
console.log(myJSON)

var myJSON = {"car1" : "black" ,"car2" : "blue"};
console.log(myJSON)
```



```
▼{car1: "black"}
   car1: "black"
   car2: "blue"
 ▶__proto__: Object
▼{car1: "black", car2: "blue"}
   car1: "black"
   car2: "blue"
 ▶__proto__: Object
▼{car1: "black"}
   car1: "black"
   car2: "blue"
 ▶__proto__: Object
▼{car1: "black", car2: "blue"}
   car1: "black"
   car2: "blue"
 ▶__proto__: Object
▼{car1: "black", car2: "blue"}
   car1: "black"
   car2: "blue"
 ▶__proto__: Object
```

Dot notation vs Bracket Notation

```
var myJSON = {}
myJSON.car1 = "black"
myJSON["car1"] = "blue"
console.log(myJSON)
```

Try It

Create an object from scratch

Output the content in the console.

Add it to your website using JavaScript.

```
<div id="output1"></div><div id="output2"></div>
<script>
    var output1 =document.getElementById('output1');
    var output2 =document.getElementById('output2');

    var myObj = { "firstName" : "Mike" ,"lastName" : "Smith" ,
"age": 30  };

    console.log(myObj);

    var name = 'Name';

    output1.innerHTML = myObj.firstName;
    output2.innerHTML = myObj['last' + name];

</script>
```

Array of items

Better way

```
var cars = {"car":["Blue","black"]}
console.log(cars)
```

Now we can add more details to each item :)

```
var myJSON = {"car1" : {"color":"black"} ,"car2" : {"color" :"blue" }};
console.log(myJSON)
```

Even more details as much as we want!!!

```
var myJSON = {"car1" : {"color":"black", "model":"Mustang"} ,"car2" : {"color" :"blue","model":"F150" }};
 console.log(myJSON)
```

Try It

Create a JSON object with more than one item. Object array.

```
<div id="output1"></div>
<div id="output2"></div>
<script>
    var output1 = document.getElementById('output1');
    var output2 = document.getElementById('output2');
    var myObj = {
        "people": [
            {
                "firstName": "Mike",
                "lastName": "Smith",
                "age": 30
            },
            {
                "firstName": "John",
                "lastName": "Jones",
                "age": 40
            }]
    };
    console.log(myObj);
    var name = 'Name';
    var i = 0;
    output1.innerHTML = myObj.people[i].firstName;
    output2.innerHTML = myObj.people[i]['last' + name];
    var i = 1;
    output1.innerHTML += myObj.people[i].firstName;
    output2.innerHTML += myObj.people[i]['last' + name];
</script>
```

Looping through Array

Loop Through the items in the object array.
Output a list of places into your HTML.

```
<div id="output1">People List<br></div>
<div id="output2"></div>
<script>
   var output1 = document.getElementById('output1');
   var output2 = document.getElementById('output2');

   for(var i=0;i<myObj.people.length;i++){
      output1.innerHTML += "<br>"+myObj.people[i].firstName
+ " " + myObj.people[i].lastName ;
    }
</script>
```

Try It

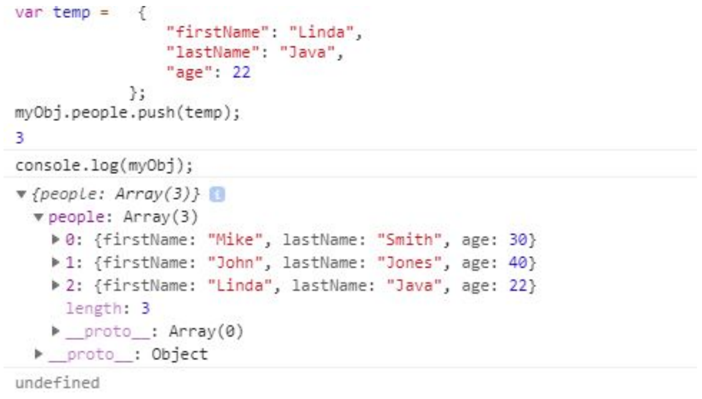
Loop Through the items in the object array. Output a list of places into your HTML.

```
var myObj = {
    "people": [
        {
            "firstName": "Mike",
            "lastName": "Smith",
            "age": 30
        },
        {
            "firstName": "John",
            "lastName": "Jones",
            "age": 40
        }],
    "places":[
        {
            "location":"Toronto",
            "lat":87,
            "long":140
        },
        {
            "location":"New York",
            "lat":67,
            "long":110
        }]
};
```

Try This

Add a new people value to the object using JavaScript. Add to the Object.

Loop through the data and output it in the page.



```
var temp =   {
                "firstName": "Linda",
                "lastName": "Java",
                "age": 22
             };
myObj.people.push(temp);
3
console.log(myObj);
▼ {people: Array(3)}
  ▼ people: Array(3)
    ▶ 0: {firstName: "Mike", lastName: "Smith", age: 30}
    ▶ 1: {firstName: "John", lastName: "Jones", age: 40}
    ▶ 2: {firstName: "Linda", lastName: "Java", age: 22}
      length: 3
    ▶ __proto__: Array(0)
  ▶ __proto__: Object
undefined
```

```
<div id="output1"></div>
<div id="output2"></div>
<script>
   var output1 = document.getElementById('output1');
   var output2 = document.getElementById('output2');
   var myObj = {
      "people": [
         {
            "firstName": "Mike",
            "lastName": "Smith",
                    "age": 30
 },
         {
            "firstName": "John",
            "lastName": "Jones",
            "age": 40
         }]
   };
   console.log(myObj);
   var name = 'Name';
   var i = 0;
   output1.innerHTML = myObj.people[i].firstName;
   output2.innerHTML = myObj.people[i]['last' + name];
   var i = 1;
   output1.innerHTML += myObj.people[i].firstName;
   output2.innerHTML += myObj.people[i]['last' + name];
</script>
```

Solution

Add a new people value to the object using JavaScript.

Loop through the data and output it in the page.

```
  var myObj = {
    "people": [
      {
        "firstName": "Mike",
        "lastName": "Smith",
              "age": 30
},
      {
        "firstName": "John",
        "lastName": "Jones",
        "age": 40
      }]
  };
  var temp =   {
        "firstName": "Linda",
        "lastName": "Java",
        "age": 22
      };
  myObj.people.push(temp);
```

JavaScript Methods

The JSON.stringify() method converts a JavaScript value to a JSON string, optionally replacing values if a replacer function is specified or optionally including only the specified properties if a replacer array is specified.

The JSON.parse() method parses a JSON string, constructing the JavaScript value or object described by the string. An optional reviver function can be provided to perform a transformation on the resulting object before it is returned.

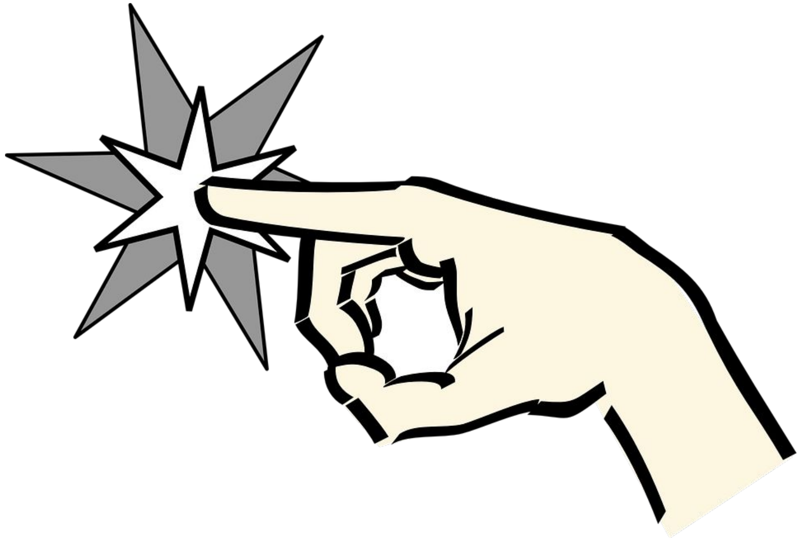
```
var json = '{"result":true, "count":42}';
var obj = JSON.parse(json);

console.log(JSON.stringify({ x: 5, y: 6 }));
// expected output: "{"x":5,"y":6}"
```

Try It JSON.parse() and JSON.stringify()

Create an object

Turn the object into a string value

Turn the string into an object



Thank you

Thank you for taking the course, and reading this PDF. If you have any questions of suggestions please connect with me on Udemy.

https://www.udemy.com/user/lars51/

Laurence Svekis