JSON basics

learn more at:
https://www.w3schools.com/js/js_json_intro.asp

JSON

JSON = Javascript Object Notation

It is a string representation of javascript data objects.

Data objects because JSON does not allow functions, or Dates for example.

Node, and Browsers, provide a JSON global function that can turn a javascript object into a JSON string:

```
   var jsonString = JSON.stringify(obj)
```

and a parser to turn a JSON string back into an object:

```
    var obj = JSON.parse(jsonString)
```

JSON

Observations:

There is no such thing as a JSON object per se. JSON is a string representation of objects. (Though people often talk about passing JSON objects around).

JSON is easy to move or store because it is just text and easy to parse with javascript's built in JSON function object.

Javascript Object vs. JSON

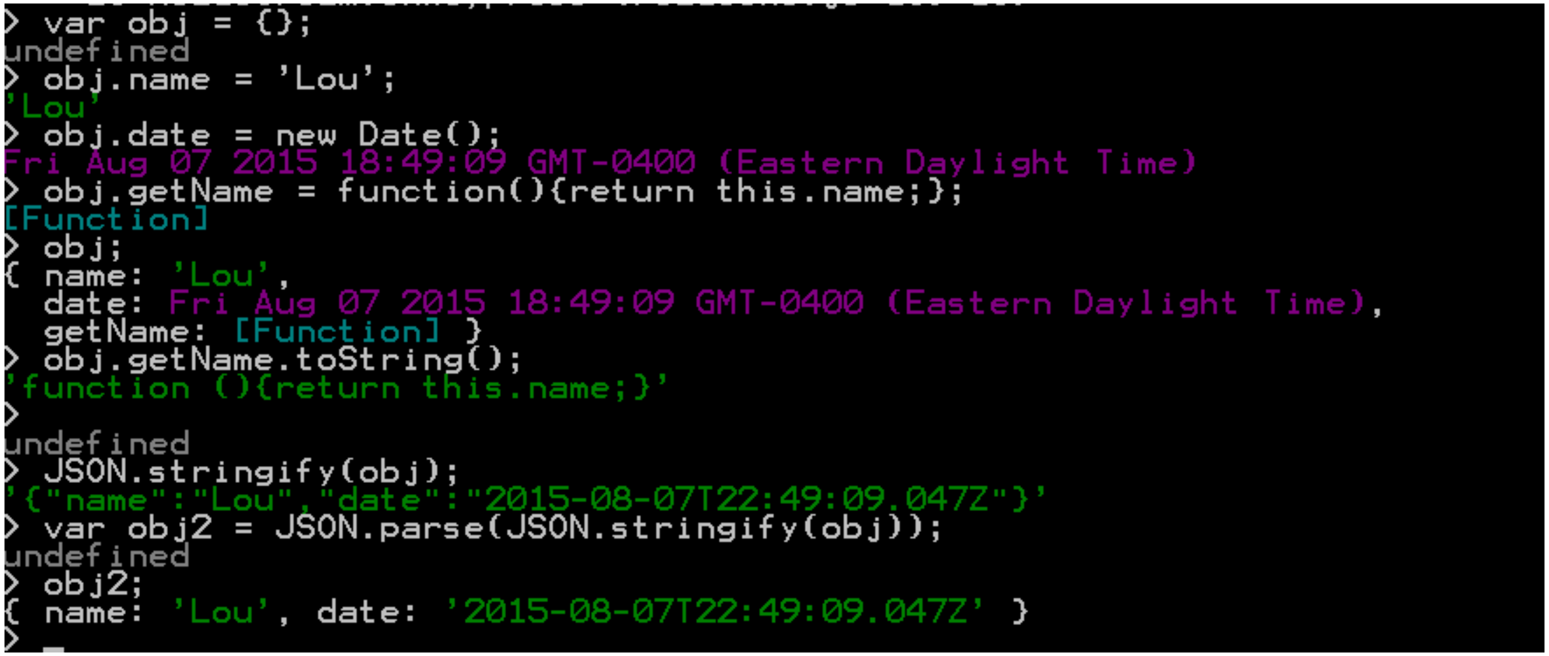
```
var obj = {
    name: 'Lou',
    date: new Date(),
    getName: function() {return this.name}
 }
```

```
 var jsonString = JSON.stringify(obj);

         '{"name":"Lou","date":"2015-08-07T22:49:09.047Z"}'
```

```
 var obj2 = JSON.parse(jsonString);

         { name: 'Lou', date: '2015-08-07T22:49:09.047Z' }
```

Observations:
When you JSON.stringify() an object, functions are ignored, dates are turned into strings.
Also, JSON strings only allow double quotes for keys, not single or double quotes the way javascript allows. (The single quote is used to delimit the overall string.)

JSON stringify() and parse() session



```
> var obj = {};
undefined
> obj.name = 'Lou';
'Lou'
> obj.date = new Date();
Fri Aug 07 2015 18:49:09 GMT-0400 (Eastern Daylight Time)
> obj.getName = function(){return this.name;};
[Function]
> obj;
{ name: 'Lou',
  date: Fri Aug 07 2015 18:49:09 GMT-0400 (Eastern Daylight Time),
  getName: [Function] }
> obj.getName.toString();
'function (){return this.name;}'
>
undefined
> JSON.stringify(obj);
'{"name":"Lou","date":"2015-08-07T22:49:09.047Z"}'
> var obj2 = JSON.parse(JSON.stringify(obj));
undefined
> obj2;
{ name: 'Lou', date: '2015-08-07T22:49:09.047Z' }
> _
```

JSON experiment.

Observations:
When you JSON.stringify() and object. Functions are ignored, dates are turned into strings.
Also, JSON strings only allow double quotes for key names (not single quotes).

Data Interchange Schemes

Three data interchange formats are the most popular with web apps:

- XML (Extendable Markup Language)
- JSON (Javascript Object Notation)
- URI Encoded Data

- All three are popular for data interchange (transmitting data through a network).

- XML and JSON both also popular as configuration data.

URI Encoded Data

```
var uri = "my test.asp?name=ståle&car=saab"
console.log(encodeURI(uri))
my%20test.asp?name=st%C3%A5le&car=saab
console.log(encodeURIComponent(uri))
my%20test.asp%3Fname%3Dst%C3%A5le%26car%3Dsaab


var enc = "my%20test.asp?name=st%C3%A5le&car=saab"
console.log(decodeURI(enc))
my test.asp?name=ståle&car=saab
enc =
"my%20test.asp%3Fname%3Dst%C3%A5le%26car%3Dsaab”
console.log(decodeURIComponent(enc))
my test.asp?name=ståle&car=saab
```

XML vs JSON

```
<?xml version="1.0" encoding="UTF-8"?>
<song>
  <title> 26-2 </title>
  <composer> John Coltrane </composer>
  <musicalStyle> Medium Up Swing
</musicalStyle>
  <key> F </key>
    <bar>
      <timeSignature> 4/4 </timeSignature>
      <rehearsalLetter> A </rehearsalLetter>
      <leftDoubleBarLine/>
      <chords> Fmaj7 / Ab7 </chords>
    </bar>
    <bar>
       <chords> Dbmaj7 / E7 </chords>
    </bar>
    <bar>
         <chords> Amaj7 / C7 </chords>
    </bar>
    <bar>
        <chords> Cm7 / F7 </chords>
    </bar>
    <bar>
      <finalBarLine/>
      <chords> Fmaj7 </chords>
     </bar>
</song>
```

```
{
  "title": "26-2",
  "composer": "John Coltrane",
  "musicalStyle": "Medium Up Swing",
  "key": "F",
    "bars": [
    {
      "timeSignature": "4/4",
      "rehearsalLetter": "A",
      "leftDoubleBarLine": "true",
      "chords": "Fmaj7 / Ab7",
    },
    {
       "chords": "Dbmaj7 / E7",
    },
    {
         "chords": "Amaj7 / C7",
    },
    {
        "chords": "Cm7 / F7",
    },
    {
      "finalBarLine": "true",
      "chords": "Fmaj7",
     }
    ]
}
```

XML vs JSON

```
<?xml version="1.0" encoding="UTF-8"?>
song>
  <title> 26-2 </title>
  <composer> John Coltrane </composer>
  <musicalStyle> Medium Up Swing
</musicalStyle>
  <key> F </key>
    <bar>
      <timeSignature> 4/4 </timeSignature>
      <rehearsalLetter> A </rehearsalLetter>
      <leftDoubleBarLine/>
      <chords> Fmaj7 / Ab7 </chords>
    </bar>
    <bar>
       <chords> Dbmaj7 / E7 </chords>
    </bar>
    <bar>
         <chords> Amaj7 / C7 </chords>
    </bar>
    <bar>
        <chords> Cm7 / F7 </chords>
    </bar>
    <bar>
      <finalBarLine/>
      <chords> Fmaj7 </chords>
     </bar>
</song>
```

```
{
  "title": "26-2",
  "composer": "John Coltrane",
  "musicalStyle": "Medium Up Swing",
  "key": "F",
    "bars": [
    {
      "timeSignature": "4/4",
      "rehearsalLetter": "A",
      "leftDoubleBarLine": "true",
      "chords": "Fmaj7 / Ab7",
    },
    {
       "chords": "Dbmaj7 / E7",
    },
    {
         "chords": "Amaj7 / C7",
    },
    {
        "chords": "Cm7 / F7",
    },
    {
      "finalBarLine": "true",
      "chords": "Fmaj7",
     }
    ]
}
```

How do we know this is a song?

How do we know this is a bar?

What is the implication of `finalBarLine: "false"`

XML vs JSON –JSON trick 1

```
song>
  <title> 26-2 </title>
  <composer> John Coltrane </composer>
  <musicalStyle> Medium Up Swing
</musicalStyle>
  <key> F </key>
    <bar>
      <timeSignature> 4/4 </timeSignature>
      <rehearsalLetter> A </rehearsalLetter>
      <leftDoubleBarLine/>
      <chords> Fmaj7 / Ab7 </chords>
    </bar>
    <bar>
       <chords> Dbmaj7 / E7 </chords>
    </bar>
    <bar>
        <chords> Amaj7 / C7 </chords>
    </bar>
    <bar>
       <chords> Cm7 / F7 </chords>
    </bar>
    <bar>
      <finalBarLine/>
      <chords> Fmaj7 </chords>
     </bar>
</song>
```

```
{"song",
 {
  "title": "26-2",
  "composer": "John Coltrane",
  "musicalStyle": "Medium Up Swing",
  "key": "F",
   "bars": [
   {
     "timeSignature": "4/4",
     "rehearsalLetter": "A",
     "leftDoubleBarLine": "true",
     "chords": "Fmaj7 / Ab7",
   },
   {
      "chords": "Dbmaj7 / E7",
   },
   {
       "chords": "Amaj7 / C7",
   },
   {
      "chords": "Cm7 / F7",
   },
   {
     "finalBarLine": "true",
     "chords": "Fmaj7",
    }
   ]
}
}
```

How do we know this is a song?

How do we know this is a bar?

What is the implication of `finalBarLine: "false"`

XML vs JSON –JSON trick 2

```
song>
  <title> 26-2 </title>
  <composer> John Coltrane </composer>
  <musicalStyle> Medium Up Swing
</musicalStyle>
  <key> F </key>
    <bar>
      <timeSignature> 4/4 </timeSignature>
      <rehearsalLetter> A </rehearsalLetter>
      <leftDoubleBarLine/>
      <chords> Fmaj7 / Ab7 </chords>
    </bar>
    <bar>
       <chords> Dbmaj7 / E7 </chords>
    </bar>
    <bar>
         <chords> Amaj7 / C7 </chords>
    </bar>
    <bar>
        <chords> Cm7 / F7 </chords>
    </bar>
    <bar>
      <finalBarLine/>
      <chords> Fmaj7 </chords>
     </bar>
</song>
```

```
{"songs": [
  {
  title: "26-2",
  composer: "John Coltrane",
  musicalStyle: "Medium Up Swing",
  key: "F",
    bars: [
    {
      timeSignature: "4/4",
      rehearsalLetter: "A",
      leftDoubleBarLine: "true",
      chords: "Fmaj7 / Ab7",
    },
    {
       chords: "Dbmaj7 / E7",
    },
    {
         chords: "Amaj7 / C7",
    },
    {
        chords: "Cm7 / F7",
    },
    {
      finalBarLine: "true",
      chords: "Fmaj7",
     }
    ]
}
]}
```

How do we know this is a song?

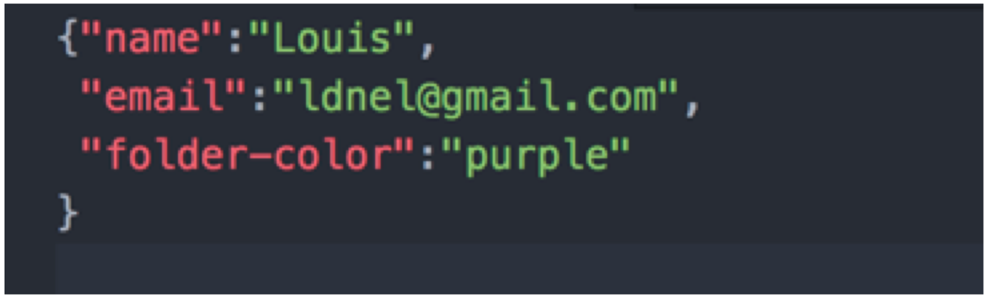
How do we know this is a bar?

What is the implication of `finalBarLine: "false"`

JSON configuration data

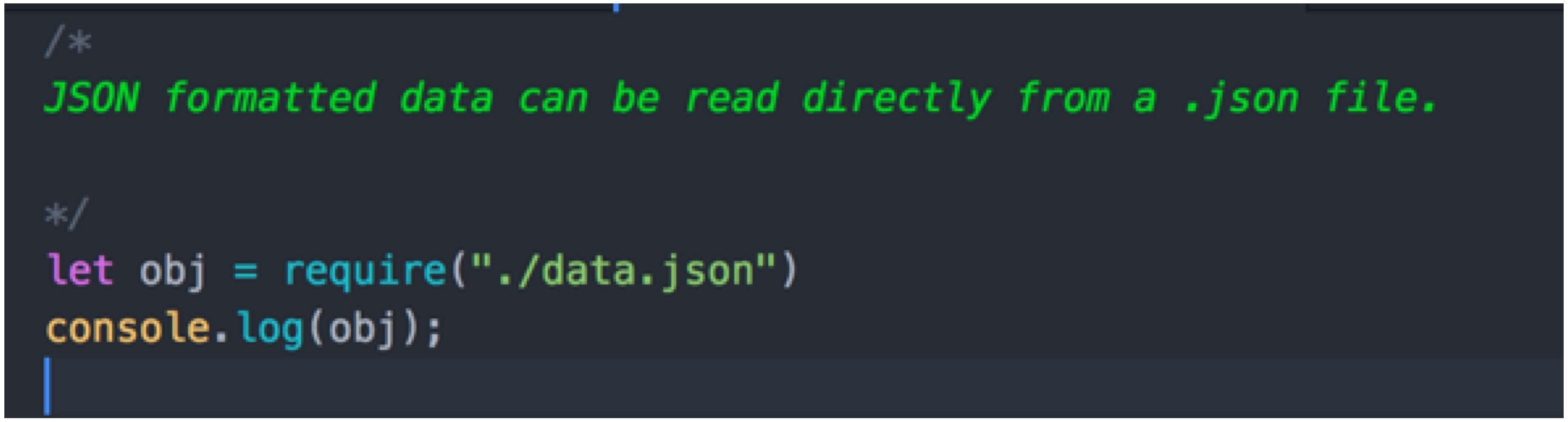
- **JSON integrates very conveniently with Javascipt.**
- **Data configuration files in JSON format can be read directly into javascript object variables using node.js `require()` mechanism**

data.json



```
{"name":"Louis",
 "email":"ldnel@gmail.com",
 "folder-color":"purple"
}
```

test.js



```
/*
JSON formatted data can be read directly from a .json file.

*/
let obj = require("./data.json")
console.log(obj);
```

JSON as interchange data format

- JSON is the primary format for moving data between web clients and javascript-based servers (like node.js-based servers)

- XML was more popular with non javascript-based servers and large application templates (e.g. MusicXML)